A Motivation-Hygiene Model of Open Source Software Code Contribution and Growth

Pratyush Nidhi Sharma
The University of Alabama
Email: pnsharma@ua.edu

Sherae L. Daniel
University of Cincinnati
Email: daniesr@ucmail.uc.edu

Tingting (Rachel) Chung
College of William & Mary
Email: rachel.chung@mason.wm.edu

Varun Grover
University of Arkansas
Email: vgrover@walton.uark.edu

Citation: Sharma, P.N., Daniel, S., Chung, T.R., and Grover, V. "A Motivation-Hygiene Model of Open Source Software Code Contribution and Growth," forthcoming in the ***Journal of the Association for Information Systems***, (2021).

A Motivation-Hygiene Model of Open Source Software Code Contribution and Growth

Abstract

The success of Open Source Software (OSS) projects depends on sustained contributions by developers who often display a wide variety of contribution patterns. Project leaders and stakeholders would strongly prefer developers to not only maintain – but preferably increase – their contributions over time as they gain experience. Corporations increasingly complement OSS developer motivations (such as fit in terms of shared values with the project community) by paying them to sustain contributions. However, practitioners argue whether payment helps or hurts projects because imbursement may dampen developer motivation in the long run. This may make it difficult for project leaders to understand what to expect from developers over time. Using Herzberg's motivation-hygiene framework, we explore how developers' perceptions of value fit with the project and being paid interact to determine the level of code contribution and its rate of change over time (i.e., growth). Using a survey of 564 developers across 431 projects on GitHub, we build a three-level growth model explaining the code contribution and its growth over a six-month period. We find that value fit with the project positively influences both the level and growth of code contribution. However, there are notable differences among paid and unpaid developers in the impact of value fit on their level and growth in code contributions over time. The implications of our work will be of interest to researchers, practitioners, and organizations investing in open source projects.

Keywords: Open source software, motivation, payment, value fit, code contribution, change over time, multilevel growth analysis

A Motivation-Hygiene Model of Open Source Software Code Contribution and Growth

Introduction

Open Source Software (OSS) developers exhibit a wide variety of code contribution patterns over time, creating challenges in anticipating the direction of project development (Wang et al. 2018). To achieve sustained development, project leaders would strongly prefer that developers not only maintain but increase their contributions over time (Ehls 2017; Lin et al. 2017). The person-organization fit literature suggests that *value fit*, which represents the extent to which an individual's personal values match the values espoused by an organization, is a strong motivating factor for long-term contributions (Moynihan and Pandey, 2008; van Viaanen et al. 2007), especially in volunteer-driven organizations (Bahat 2020; Ertas 2019). OSS literature also suggests that developers are driven to contribute based on the shared values espoused by a project community (Schilling et al. 2012; Shah 2006; Stewart and Gosain 2006).

Although OSS development is based on certain core values (Raymond 2001), project communities and individual developers differ widely in how strongly they embrace them. While some projects welcome corporate sponsorship and choose permissive licenses (e.g., MIT license), others shun sponsorship and choose restrictive licenses (e.g., GNU GPL) (Stewart et al. 2006; Ho and Rai 2017). Developers also vary in how strongly they believe in the OSS values of altruistic software sharing, use and reciprocation (Maruping et al. 2019). Thus, it is not merely how deeply developers hold these values but rather the extent to which they match with a specific project community that will drive their contributions (Kristoff 1996).

Increasingly, many developers also rely on their OSS work to earn money to support their livelihoods (Kantrowitz 2015; Schlueter 2013). Many firms (e.g., Microsoft, Google) now pay select developers to support and sustain their OSS contributions (Germonprez et al. 2013; Riehle et al. 2014). However, the introduction of payment is considered by many as being at odds with the OSS philosophy and community values (Fitzgerald 2006; Gerlach 2016). On the one hand, some argue that not paying developers for their labor is unethical and prevents many talented developers from contributing (Dryden 2013; Werdmuller 2017). In the worst case, critical software defects may remain hidden or unaddressed due to lack of paid developers, as evidenced by the Heartbleed flaw in OpenSSL (Brodkin 2014; Marquess 2014).

On the other hand, others point to the uncertainty payment generates regarding the shared values, effort expectations, and the future of the project development because paid and unpaid developers often have different priorities (Berdou 2011). For example, David Hansson, the creator of Ruby on Rails, argues that payment can demotivate OSS developers, rein in their creativity, and disrupt the core community values in the long run (Hansson 2013). The Debian community, which recently experienced extensive disruption in development, offers a cautionary tale in this regard where many members argued that payment was against the values espoused by the project (Gerlach et al. 2016). Based on this ongoing concern, our research question is: *How does the interaction between developers' perceptions of value fit with an OSS project and payment impact their code contribution levels and its rate of change over time (i.e., growth)?*

We utilize Herzberg's motivation-hygiene theory to illuminate the factors that lead to developers' code contribution patterns over time (Herzberg 1968; Herzberg et al. 1966). Herzberg suggests that factors facilitating individuals' psychological growth (i.e., motivation factors) distinctly impact outcomes compared to those that affect their physiological needs (i.e., hygiene factors). Specifically, we posit that developers' perceptions of value fit with the project functions as a motivation factor, while payment serves as a hygiene factor. Even though value fit is an important motivator in OSS development (Maruping et al. 2019), it can clash with the influence of firms paying developers (Kreutzer and Jäger 2011). Research in psychology suggests that external rewards, such as financial remuneration, may dampen developer motivation (Deci et al. 1999; Kohn 1999). Yet, Roberts et al. (2006) found no evidence of diminished motivation in the presence of financial motivators for OSS developers. By considering the longitudinal impact of time, we seek to resolve the apparent discrepancy in these results in the context of OSS development. In particular, exploring the tension between value fit and payment can reveal the limits of their ability to elicit code contributions over the long-term.

To do so, we build a three-level growth model within the hierarchical generalized linear modeling (HGLM) framework on a sample of 564 developers working on the GitHub platform across 431 projects. We test our hypotheses predicting code contribution level and its growth over a six-month period as a function of developers' perceptions of value fit with the project, and how payment moderates this effect. We find that value fit has an overall positive influence on the level of code contribution and its growth. However, important differences exist in the effect of value fit between paid and unpaid developers on their level of code contribution and growth. Specifically, value fit has a stronger effect on the level of code contribution for paid developers compared to unpaid developers. In contrast, value fit has a stronger effect on the growth in code contribution over time for unpaid developers compared to paid developers. We discuss the implications of our work for the long-term outlook of developer contributions in OSS projects.

Background

OSS Developer Code Contribution Patterns

Developer code contributions—in the form of new features, enhancements, and bug fixes via code commits—constitute the core activity of OSS communities to ensure the long term viability of the software product (Crowston et al. 2012). However, team stability is difficult to achieve in OSS projects because developers display a wide variety of code contribution patterns over time (Wang et al. 2018). Many volunteer developers make a single contribution without ever making another one (Lee 2018), while others participate infrequently over the long term (Barcomb et al. 2018). Only a limited set of developers make sustained contributions to a particular project (Fang and Neufeld 2009; Pinto et al. 2016; Qureshi and Fang 2010).

Research has documented the severe negative effects of turnover among software developers in general, and paid OSS developers in particular on project development (e.g., Foucault et al. 2015; Lin et al. 2017). The effect of lack of long-term developer contributions can be especially severe for OSS projects due to the absence of general training or formal onboarding procedures to bring new developers up to speed (Robles and Gonzalez-Barahona 2003). Both paid and volunteer developers face social and technical contribution barriers in acclimatizing to the complex project environment and must invest significant effort and time before being allowed to join the "core" group, making team regeneration very challenging (Steinmacher et al. 2015; Von Krogh et al. 2003). Not surprisingly, understanding factors that influence OSS code contribution patterns has attracted significant research attention (Von Krogh et al. 2012).

Motivation-Hygiene Theory

Herzberg's motivation-hygiene theory addresses how an organization or collective can effectively motivate employees or members to be productive. It proposes that human beings experience two fundamental drives: to attain maximum psychological growth and to avoid pain. Herzberg labeled these two distinct sets of factors *motivation* and *hygiene* (Herzberg 1968; Herzberg et al. 1966). He argued that psychological growth factors - such as the inherent purpose of work gained by collaborating with others who hold similar values - serve as motivators; while hygiene factors, such as salary, do not motivate, but instead act as de-motivators when perceived negatively (Katt and Condly 2009). The motivation factors tap into relational and emotional issues, while the hygiene factors align with economic and utilization needs.

Herzberg developed the idea of two separate factors based on his work in the health and epidemiology fields. When he asked employees about the best aspects of their jobs, they mentioned interesting work and their interrelationships with peers who shared their values. He termed these factors “motivators” because they are essential for intrinsically motivating individuals (Deci et al. 1999). On the other hand, when asked about the worst aspects of their jobs they discussed factors like pay, which he found were related to employees’ dissatisfaction, but not their satisfaction (Herzberg et al. 1966; Sachau 2007). Herzberg borrowed the term “hygiene” to describe these factors because good hygiene prevents illness, but it does not make one healthy. Herzberg’s hygiene factors help meet the basic needs to sustain decent livelihoods, and influence job dissatisfaction, but on the contrary, have minimal impact on satisfaction. Payment for work, for instance, holds more potency as a job dissatisfier than as a job satisfier, and is a necessary but not sufficient condition for satisfaction (Sachau 2007). Individuals driven by hygiene factors may not necessarily find pleasure in doing the task (such as writing open source code), but the financial remuneration keeps them from getting frustrated by helping support their livelihoods. This occurs because the opposite of satisfaction is not dissatisfaction, but “no satisfaction.” That is, satisfaction and dissatisfaction do not represent opposite ends of the same spectrum, rather they represent separate continuums. Employees who find value in their work may find their motivation frustrated by the absence of hygiene factors (e.g., payment). Furthermore, employees who get their financial needs met can still lack motivation because their jobs do not offer any value. Accordingly, Herzberg suggests providing motivation factors and attending to hygiene factors at the same time in order to maintain a motivated workforce – but keeping their administration separate (Herzberg 1968; Sachau 2007).

Longitudinally, the theory holds that motivation and hygiene factors have distinct impacts on satisfaction versus dissatisfaction (or frustration) over the short and long-terms. Motivation factors facilitate an individual's psychological growth and lead to long-term satisfaction (Herzberg 1968; Sachau 2007). On the contrary, hygiene factors typically yield short-term or temporary "highs" when administered but lead to immediate frustration when withheld causing the individual to likely stop or ignore work (Herzberg 1965; Herzberg 1968). This is supported by research that shows that wealth and payment do not lead to long-term satisfaction but can prevent immediate frustration (Sachau 2007). For example, in a study across 40 countries, Inglehart (1997) found that as long as the basic needs of life are met, wealth (or income) does not lead to long-term satisfaction. Large meta-analyses of studies also consistently show that payment for work is unrelated to long-term satisfaction (Haring et al. 1984). Even windfalls in cash, such as lottery, do not have long-term positive impacts (Brickman et al. 1978). Thus, hygiene factors are most effective at preventing immediate suffering rather than bringing long-term happiness or satisfaction (Sachau 2007). Instead, motivation factors are considered more effective in driving the individual to superior performance and effort over time. Indeed, Herzberg (1968, p. 62) notes, "The very nature of motivators, as opposed to hygiene factors, is that they have a much longer-term effect on employees' attitudes". Next, we discuss the two main motivations that underlie Herzberg's theory in the context of OSS development.

Motivation Factor: Value Fit

The desire to be with similar others is innate in human beings and motivates them to seek situations where this need can be met (Schneider 1987). We posit that *value fit*, which is the condition where OSS developers' drive to find meaning in their work flourish in a *given* project while collaborating with others with shared values, acts as a motivation factor (Kristof 1996; Schilling et al. 2012). Value fit is most frequently operationalized as the congruence between organizational and individual values, norms, and beliefs (Kristof 1996; Moynihan and Pandey, 2008; van Viaanen et al. 2007). Personal values, norms, and beliefs are enduring, deeply embedded motivational resources that guide people in associating with others like themselves and are separate from their day-to-day needs (Schwartz 1992). Recent work on social networks and online collaboration shows that people are motivated to associate with similar others who reaffirm, rather than challenge, their values and beliefs (Boutyline and Willer 2017). Indeed, the desire to "fit in" and partake in the values of the hacker culture (e.g., freedom of software sharing and use, reciprocation) was the main reason behind the open source movement (Raymond 2001). Working, interacting, and most importantly, being accepted by reciprocating others with similar values, and thus successfully fitting in the culture, gives OSS developers a sense of purpose and leads to the creation of strong in-group identities (Lakhani and Wolf 2005).

The major mantras in OSS practice have been valuing knowledge, code sharing and use, community governance, and volunteer work – regardless of the financial incentives (Ke and Zhang 2009; O'Mahony and Ferraro 2007; Stewart and Gosain 2006). In addition, projects differ in terms of community norms that reflect ways of practicing software development around issues related to reciprocation and offering named credit to deserving developers to value their work (Stewart and Gosain 2006; Maruping et al. 2019). Similarly, community norms related to project forking are an important component of value fit between a developer and project community.[1] While some developers and communities consider forking a cardinal sin because of its negative impact on the project, others developers, including Brian Behlendorf, the co-founder of Apache

[1] Project forking happens when a project community and its resources are split into two or more streams for independent development.

Software Foundation, view project forking as a valid response against governance disagreements to create new communities with shared values and sustainable development (Gamalielsson and Lundell 2014). For example, LibreOffice is a fork of the OpenOffice project created by members who hold more egalitarian values. Thus, the match between a developer and a project community regarding named credit, forking, reciprocation, and beliefs regarding OSS practice positively influences continued participation (Shah 2006; Stewart and Gosain 2006).

Individual projects also provide signals to potential developers about their community values. Specifically, license choice is an important means by which a project communicates its values regarding code sharing and profit making to developers (Spaeth et al. 2014; Stewart and Gosain 2006). For example, the GNU "copyleft" license embodies the political value and belief system espoused by its progenitor, Richard Stallman (Brock 2013). Indeed, Stallman writes:[2]

> "*Every decision a person makes stems from the person's values and goals. People can have many different goals and values…When the goal is a matter of principle, we call that idealism. My work on free software is motivated by an idealistic goal: spreading freedom and cooperation. I want to encourage free software to spread, replacing proprietary software that forbids cooperation, and thus make our society better. That's the basic reason why the GNU General Public License is written the way it is—as a copyleft.*"

While GNU public license is considered restrictive (Stewart et al. 2006), OSS projects can choose a permissive and business-friendly license that better reflects their values (Lerner and Tirole 2005). When a project selects a license, it sends a signal about its community values with which a developer may or may not agree. Similarly, a project's choice of firm sponsorship sends another signal to developers regarding their fit with the community. Stewart et al. (2006) found that firm sponsorship discourages developers who value autonomy and disdain profit motives. For example, while some developers are supportive of sponsorships, other prominent developers

[2] https://www.gnu.org/philosophy/pragmatic.html

such as David Hansson consider it a "grave risk to the culture of open source".[3] Thus, developers' perceptions of fit regarding whether organizational sponsorship is acceptable and by whom can determine their motivations to contribute (Spaeth et al. 2014). To the extent that developers perceive a match with the project community along these characteristics, they will feel a sense of belonging that drives positive outcomes (Kristof-Brown and Guay 2011). In particular, value fit can be a crucial factor in determining developers' retention in the project when their learning and skill development eventually plateaus out (Zhou et al. 2016).[4]

Hygiene Factor – Payment in OSS

Consistent with Herzberg et al. (1966), we argue that payment for OSS work acts as a hygiene factor. While in its traditional avatar OSS development mostly relied on unpaid volunteers, a significant number of developers now receive payment and work full time on OSS projects to support themselves financially (Germonprez et al. 2013; Riehle et al. 2014). For example, Hertel et al. (2003) reported that 20% of developers on Linux were paid to contribute on a regular basis and 23% more were paid to contribute occasionally. More recently, Corbet et al. (2012) found that successful OSS projects received more than 75% of their code from developers who received payment from a company.[5] In fact, commercial developers support

---

[3] https://twitter.com/dhh/status/1131585498395242496

[4] In contrast to value fit, which is "supplementary" in nature, two types of "complementary" person-organization fit also exist that measure the extent to which individuals and organizations are able to provide what is missing in the other (Kristoff-Brown et al. 2005). First, Need-Supply fit measures the match between an individual's pragmatic needs and an organization's ability to satisfy them. OSS developers also contribute to projects to fulfil a variety of personal needs (need for software, enhancing their careers, enjoyment, learning etc.) that are separate from their value-based motivations (Shah 2006). Second, and reciprocally, Demand-Ability fit measures the match between an organization's demands and an individual's ability to successfully meet them. OSS projects place specific demands on developers in terms of their knowledge, skills and abilities when making technical contributions such as code commits. We discuss the important role of these two complementary fit types as covariates in our model later.

[5] Financial incentives given to OSS developers may take a variety of forms. In addition to salary, some developers receive bounties, donations or grants (Krishnamurthy and Tripathi 2009). A few OSS communities fundraise in order to pay select developers; for instance, members of the Debian community raised money to pay two release managers (Gerlach et al. 2016). Companies also pay developers to contribute to OSS projects to build their technical expertise, which can then be utilized in their business models. For example, RedHat and MySQL pay employees to write code to gain expertise so that they can help in product maintenance, support installation. Other for-profit

embedded Linux more so than hobbyists (Henkel 2006). Increasingly, many volunteer OSS developers are seeking payment for their work to support their livelihoods. For example, Isaac Schlueter, the creator of the *npm* package manager for Javascript notes (Schlueter 2013):

> *"Life costs money. The de facto way to get that money is to have a job. Some forego the corporate gig, in favor of being nomads and starving artists. They take on minimal employment requirements, if any, and spend the rest of their time being productive on open source. But it's a rough way to live. Good luck sending a kid to school that way, or even feeding one. If we are going to continue to get benefits from Open Source Software, and especially if we are going to maximize those benefits further, we have to figure out how to pay for it. Beyond enabling OSS developers to eat and live indoors, payment ties our efforts to the "real world" of transactions, where people use our software to do stuff. Otherwise, it's all too easy to spiral off into ivory tower la la land."*

In contrast to traditional paid software development work, and despite its benefits to OSS developers, introducing financial incentives in OSS communities can be challenging due to its potential to create complex feelings among OSS developers. Critics argue that paying OSS developers damages their creativity, motivation, and "*risks transporting a community of peers into a transactional terminal. And that buyer-seller frame detracts from the magic that is peer-collaborators. It also holds the threat of corrupting the community at large*" (Hansson, 2013). Paying developers also increases complexity associated with governance and collaboration (Jensen and Scacchi 2005), and developers worry that firms who pay may take control over the OSS project (Gerlach et al. 2016). Developers on the Debian project expressed negative emotion because they felt payment went against the project's espoused values, in addition to creating inequity among developers (Gerlach et al. 2016). In spite of these issues, not receiving pay for their work to support their livelihoods can frustrate OSS developers and affect their contributions (Dryden 2013; Marquess 2014; Werdmuller 2017). Thus, payment can generate complicated issues in the OSS context that may affect developers' long-term attitudes in unpredictable ways.

---

companies sell applications that they build on top of OSS products.

The Role of Time

Time plays a central role in the motivation-hygiene theory. Herzberg suggests that motivation factors drive long-term impacts, while hygiene factors have short-term or temporary impacts (Herzberg 1968; Herzberg et al. 1966; Sachau 2007). The OSS literature also suggests time as the key factor in the relationship between developer attitudes and behaviors. Developers initially face significant social and technical barriers to experiencing the benefits of value fit and contributing to OSS projects (Steinmacher et al. 2015). Because OSS projects often do not offer formal onboarding processes or training, a developer interested in joining can initially learn about their possibility of fitting in by reading project-related documentation. Projects typically maintain forum posts, licenses, sponsorship information, web pages, code documentation, or FAQs that provide an immediate, yet somewhat limited insight into their potential value fit.

Over time, the experiences interacting with other developers reinforce whether or not a developer fits with the project community values (Schneider 1987). They can glean signals regarding whether the fit exists from their social interactions with other developers, the type of contributions that the community prizes, and the direction of the project's future development (Ducheneaut 2005). Interaction with project members reduces developer's uncertainty regarding the contribution process, facilitates mutual expectations and workflow, and the achievement of individual and group goals. Developers can learn and acquire the social and technical knowledge necessary over time, thereby leveraging their value fit with the project further to contribute successfully (Ducheneaut 2005; Qureshi and Fang 2010). They learn about the project "culture," including its values for meaningful social interaction and collaboration, before being accepted by others. The degree to which they are successful in doing so represents the strength of their value fit with the project (French et al. 1974). Thus, the benefits of value fit compound over time. In contrast, a lack of value fit will lead to developer withdrawal (Schneider 1987). Overall, the short-term impacts of motivating factors like value fit are different from the long-term impacts.

Interaction between Motivation and Hygiene Factors

What happens to developers' attitudes about contributing over the long-term when they experience both hygiene and motivation factors at the same time? The literature offers mixed evidence. On the one hand, vast literature in psychology shows that extrinsic rewards, which are analogous to Herzberg's hygiene factors, weaken the positive impact of intrinsic motivations, which are analogous to Herzberg's motivation factors (Deci et al. 1999; Frey and Jegen 2001; Kohn 1999; Sachau 2007; Wiersma 1992). When a developer receives compensation for contributing code, the power structure within the OSS community changes (Jensen and Scacchi 2007). Instead of the developer holding autonomy in how and when to contribute, the power to determine their actions moves to their employer or sponsor. The employers may mandate the developer to perform tasks that may not hold their interest, thereby reducing their autonomy and motivation (Atiq and Tripathi 2014). This is problematic because people have an inherent need to feel in control of their actions and be autonomous (Ryan and Deci 2000).

On the other hand, a recent stream of OSS literature disputes the findings above and suggests that receiving payment does not dampen developer motivation (Alexy and Leitner 2011). Researchers found that financial rewards held no negative impact on developer motivation in the OSS context (Hars and Ou 2002; Lakhani and Wolf 2005; Roberts et al. 2006). In fact, Lakhani and Wolf (2005) found that receiving financial rewards led developers to contribute more time to OSS development. Similarly, Roberts et al. (2006) argued that developers who receive payment have their motivations complemented because they find more time to work on projects they are interested in. In contrast, Herzberg argues that hygiene factors yield benefits only up to a certain extent, beyond which an organization must provide other motivators to keep employees contributing (Sachau 2007). Given the uncertainty in how the motivation and hygiene factors interact over time we develop our research hypotheses next.

Research Model and Hypotheses

Effect of Value Fit on Code Contribution

Interacting with similar others fulfils people's fundamental need for validation of their perspectives and finding meaning in their work (Byrne 1961). Because personal values and beliefs often drive OSS developers (Bagozzi and Dholakia 2006; Gerlach et al. 2016; Ljungberg 2000), value fit with a project and its unique environment offers a way for them to meet their personal need for consensual validation (Kristof-Brown et al. 2005). Given the wide variety of developer and project value systems, it is the congruence between them that determines whether developers will perceive value fit, experience opportunities for psychological growth, and therefore continue to contribute to projects (Kristof 1996; Schneider 1987). Value fit with a project reinforces and validates their values and beliefs, thereby increasing their commitment to it (Edwards and Cable 2009; Moynihan and Pandey 2008; van Viaanen et al. 2007).

Fitting in also brings other advantages to developers that facilitate their code contributions. Value fit with the project reduces their contribution cost by lowering the interpersonal and communication barriers with others, clarifying expectations, streamlining the effort, and informing them about how to make valuable contributions (Kristoff-Brown et al. 2005; Maruping et al. 2019). This helps meet their inherent need for competence, autonomy, and relatedness (Deci et al. 1999). For instance, due to clearer expectations they feel more confident to undertake tasks they know others will value. Due to low interpersonal and communication barriers, they can better relate with their peers. Together these factors allow them to contribute successfully to the project (Maruping et al. 2019). In the process, they may extract benefits including opportunities to learn, earn rewards, and peer recognition (Atiq and Tripathi 2014). When a person perceives value fit with the environment, they feel a sense of positive affect, such as enhanced motivation to contribute (Kristoff-Brown et al. 2005). The degree to which the developers perceive value fit should determine the extent of their motivation to contribute to the project (Schilling et al. 2012; Schneider 1987). Therefore, as shown in Figure 1, we propose:

H1a: Value fit with the project positively affects developer code contribution.

By interacting with others and working in the project over time, developers encounter more opportunities to assess their value fit with the community (Moreland 1999). Both the project community and the developer exert reciprocal influence on one another and experience important temporal changes in their interaction over time (Moreland and Levine 1982). When the developer engages in an ongoing evaluation of the rewarding nature of their relationship with the project, their motivation to contribute increases. Indeed, motivation factors (e.g., the value and meaning of work) positively influence employees for a relatively long time (Sachau 2007).

As long as developers continue to perceive value fit with the project, they will continue their association with it, resulting in improved outcomes over time (Kristof 1996; Schneider 1987). Developers' increasing experience further improves performance as they explore, refine, and replicate new routines or tasks for performance improvement, while getting better at executing existing ones (Huckman et al. 2009). Their interpersonal interactions with other developers help them learn how to successfully navigate the social collaboration process and develop shared mental models over time (Moreland 1999). Through the process of cognitive sense-making and uncertainty reduction over a period of time, developers improve at writing code and expand their knowledge of project architecture, further reducing the technical barriers and cognitive effort required to contribute (Von Krogh et al. 2003). As their peers come to appreciate their contributions, developers may feel also psychologically rewarded by helping others, thereby providing meaning to their work and further enhancing their motivation. Because value fit lowers the contribution costs and increases the benefits accrued over time, developers can make more frequent contributions with reduced effort (Kristof-Brown et al. 2005). Thus,

H1b: Value fit with the project positively affects the growth in developer code contribution over time.

Interaction between Value Fit and Payment

OSS developers may be driven to contribute by value fit and payment simultaneously (Roberts et al. 2006). Unlike value fit that drives individuals to action by motivating them over the long-term, hygiene factors, such as payment, change individual's immediate actions because they push them not to lose those benefits (Herzberg et al. 1966). In this sense, developers contribute to the project to avoid the discomfort of losing their remuneration. In the short-term, payment keeps developers from getting immediately frustrated and dissatisfied (Sachau 2007). Indeed, paid developers report that receiving payment drives them to spend more time working on OSS than their peers (Germonprez et al. 2016; Lakhani and Wolf 2005). More importantly, their work contracts may require them to work specific hours on OSS projects resulting in a constant stream of contributions. For example, paid developers made higher than average code contributions than unpaid volunteers did in the Apache project (Roberts et al. 2006).

When developers receive payment for their contributions and simultaneously perceive value fit with the project community, they enjoy two complementary benefits. According to Herzberg et al. (1966), payment ensures that they are not immediately frustrated or dissatisfied, while the value fit makes them feel satisfied. Their need for consensual validation of perspectives is met because of their value fit with the project, at the same time they get their immediate physiological hygiene needs met by being paid (Sachau 2007). Organizational studies show that workers are most effective when they are driven by both motivation and hygiene factors simultaneously (Katt and Condly 2009). Thus, the net effect of payment in the short term is to lift the amount of motivationally (value fit) driven contributions by paid developers higher than the unpaid developers. Therefore, we propose that the level of code contribution is highest when a developer perceives value fit with the project and receives payment at the same time.

H2a: The positive effect of value fit on code contribution is stronger for paid than unpaid developers.

In contrast to our previous hypothesis, here we argue that the impact of value fit on the *growth* in code contribution activity is stronger for unpaid compared to paid developers. That is, we expect a stronger increase in code contributions over time for unpaid compared to paid developers. Our logic rests on the asymmetry in the temporal effects of hygiene and motivating factors in the long-term, and the differences in the cumulative benefits enjoyed by unpaid developers versus the unique constraints and conflicts faced by paid developers that inhibit them from fully leveraging the motivating effects of value fit over time. In essence, unpaid developers rely on their autonomy to better leverage their value fit and improve their code contribution rates over time than the paid developers.

Hygiene factors (payment) have short-term positive effects that decline with time, while motivational factors (value fit) have long-term positive effects (Sachau 2007). Herzberg (1968) found that payment produced positive feelings in a sample of accountants and engineers, but this effect was short lived. He referred to this as a “kick in the pants” that propelled employees in the short-term, but its effect diminished with time as they got accustomed to the remuneration, requiring greater amounts to get the same level of output. In other words, the potency of payment to reduce dissatisfaction weakens over time, thereby negatively impacting contribution levels. Because hygiene needs escalate while motivators do not (Sachau 2007), Herzberg argued that this can even impede the positive effects of motivators such as value fit in the long-term. Thus, while payment pushes developers to increase their contribution levels initially, its positive effect wears off over time (Sachau 2007).

In addition, contractual constraints also inhibit the self-actualization, psychological growth, and the associated growth in code contribution for paid developers over time (Ryan and Deci 2000). Due to their obligations, paid developers must prioritize their employer's vision at the cost of personal interests. They must manage intellectual property in a strategic way that benefits their employer, such as by making sure that they do not include proprietary code in the OSS application, which increases their burden (Henkel 2008). They also must carefully manage their interactions within the OSS community, lest they provide unintended benefits to competing companies, or go against project's values (Germonprez et al. 2016). As opposed to unpaid developers who can creatively contribute and are free from these burdens, paid developers may struggle to find a good balance between corporate managerialism and voluntary motivation (Kreutzer and Jäger 2011), and find it difficult to increase their rate of contribution despite their increasing familiarity with the project and its code architecture.

Over time paid developers may also feel conflicted due to employer priorities that may contradict their or project's values (Berdou 2011), inhibiting their ability to fully leverage their value fit. Reconciling the employer's strategic vision with the values that the developer and the project community share may require extra effort. Paid developers who share the project's values that all code should be open source may feel conflicted if their employer desires to keep some code private and proprietary (Bergquist et al. 2009). For example, Henkel (2006) found that firms operating in the embedded Linux market strategically limited their community participation by 'selectively revealing' software code to protect their interests. This creates hurdles and dampens the motivation for paid developers' participation in the OSS community which believes in reciprocation and sharing (Henkel 2008). Payment also moves the locus of control away from developers to their employers. This reduction in autonomy means that paid developers may not be able to choose the tasks they find personally fulfilling, challenging, and enjoyable, limiting their self-actualization and satisfaction (Lakhani and Wolf 2005; Ryan and Deci 2000). Instead, their employer can require them to work on mundane tasks, such as maintenance, that may dampen their motivation to contribute (Berdou 2011). Indeed, as Hansson (2013) notes, "*There is something lost when you share because you must, rather than because you can. It's also what leads to consultant-ware (software that's needlessly complex, and requires you to buy consultants to figure it out)*." Paid developers also have limited flexibility and freedom to choose other developers they enjoy working with, inhibiting their ability to nurture strong interpersonal bonds (Xu et al. 2009).

In contrast, unpaid developers enjoy a unique situation in that they have the freedom to be creative and act without constraints imposed by an employer, allowing them to choose the type of code contributions that best meet their psychological needs. Having sufficient leeway to develop and implement their ideas, flexibility in choosing when and what to work on, and yet possessing the ability to shape and influence the project can be very motivating for unpaid volunteers (Kreutzer and Jäger 2011). Initial value fit assessment may require unpaid developers to spend time and effort up front, possibly resulting in slower contribution rate to begin with. However, over time as they gain technical expertise and overcome the contribution barrier, they should see a faster increase in their code contribution levels. They have the freedom to choose coding tasks that they personally feel facilitate learning and enhance personal growth without worrying over imposed deadlines from their employer (Shah 2006). As they become deeply involved in the project, over time they can realize even greater benefits related to the fun in programming, learning, and reputation, while leveraging their gained skills and knowledge of the project (Shah 2006). Additionally, unpaid developers are free to interact with other developers they perceive as worthy of their attention, further strengthening the interpersonal bonds they nurture over time. Indeed, Hahn et al. (2008) demonstrate that developers strongly prefer to work with others with whom they share positive past experiences. Working with preferred collaborators enhances their sense of community (Oh et al. 2016), self-actualization (Ryan and Deci 2000), and psychological growth (Herzberg 1968), thereby allowing them to strongly leverage the motivating effects of value fit for their contributions over time. Thus, we propose,

H2b: The positive effect of value fit on the growth in code contribution over time is stronger for unpaid than paid developers.

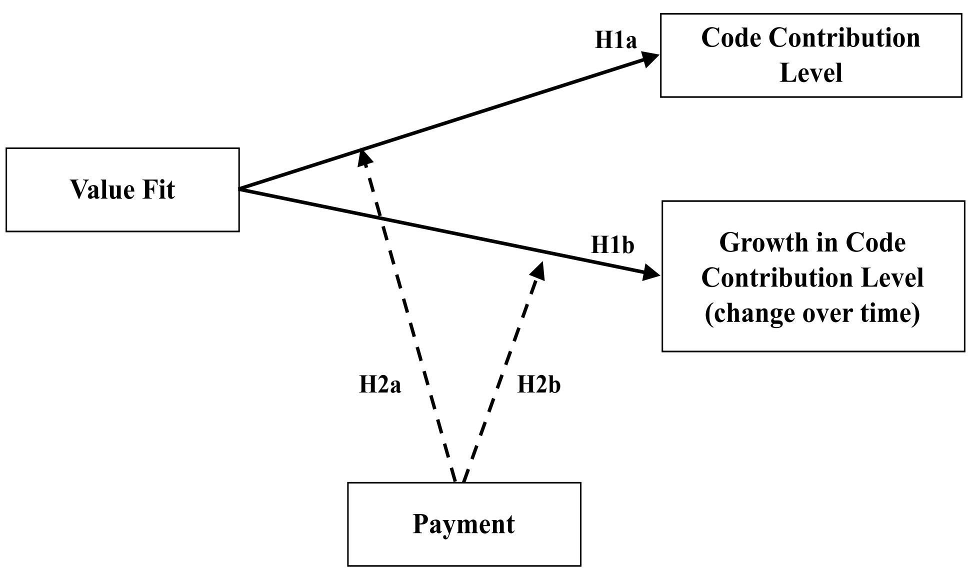

Figure 1: Research Model

Methodology

Data and Sample

We use both survey and archival data to explore the interaction between developers' perceptions of value fit and payment on their code contribution and its growth. The survey data included developer demographics, measures for value fit, and whether they were paid to work on the project. Archival data captured the pattern of their actual code contributions and other characteristics, in addition to project characteristics. We conducted a survey of developers from GitHub (www.github.com), which is one of the largest open-source platforms that provides OSS developers with communication tools, version control processes, and repositories to manage their code development. GitHub data have been widely used in OSS research (e.g., Jarczyk et al. 2018; Palyart et al. 2017). A random sample of 2,379 active developers, out of over 3 million registered on GitHub, was drawn based on the selection criteria that required developers to have made at least three code commits to a project with at least three other active developers. This was done to ensure that the projects were active and that there were enough team members to render the notion of value fit important. We approached developers through email for the survey and offered a random drawing for prizes to incentivize participation. The prizes were one Linux laptop and ten other respondents received Amazon gift cards worth $50.

A total of 768 participants responded to the survey (32.28% response rate), out of which 564 provided complete responses and were selected for the study. Respondent demographics are presented in Table 1. These developers were associated with 431 unique projects, thereby ensuring a broad range and diversity in project environments to be considered (Kristof-Brown et al. 2005). A *t*-test indicated no significant differences between respondents and non-respondents in their overall commits, recent commits, the number of developers who they followed, or the number of projects to which they contributed, indicating that response bias was not a concern. To eliminate common method bias we randomized the order of the survey questions and triangulated the criterion data from GitHub archive to separate the measurement source of predictors and outcomes (Podsakoff et al. 2003).

Table 1: Respondent Demographics

		N	Percent
Gender	Male	541	95.92
	Female	22	4.07
Education	Some High School	17	3.01
	High School Diploma	34	6.03
	Post High School	10	1.77
	College but No Degree	93	16.49
	College Degree	208	36.88
	Graduate Training	31	5.50
	Master's or Higher	171	30.32
Employment Status	Not Employed	45	7.98
	Employed in Private/Public/NGO	388	68.79
	Independent/Freelance	66	11.70
	Business Founder	38	6.74
	Self Employed	11	1.95
	Other	16	2.84

Measures

Value fit with the focal project (predictor variable): *Value Fit* measures the match between a developer and project in terms of values, beliefs, and norms (Cable and DeRue 2002). The OSS specific content dimensions along which *Value Fit* is measured were culled from the existing OSS literature (Stewart et al. 2006; Shah 2006; Stewart and Gosain 2006). We used direct measures that involve asking people whether they believe a good fit exists along a specific dimension, e.g., a project's license choice (Cable and DeRue 2002). Compared to other (indirect) measures, direct measures have been consistently shown to have the largest effect when the construct is "perceived" value fit, i.e., when fit is conceptualized as the judgment that a person fits well with the values in the organization (Kristof 1996). Furthermore, it has been shown that it is not the actual rather the perceived value fit that best predicts individual outcomes, i.e., value fit exists as long as it is perceived to exist (Cable and DeRue 2002; Lauver and Kristof-Brown 2001). Appendix-A presents the survey items used.

Payment in the focal project (predictor variable): We asked each survey respondent whether they received any form of financial compensation (e.g., salary, contract) for their participation in the focal project. The binary variable *Paid* captures this in our model.

Month (trajectory variable): We added the time-varying *Month* variable to assess the trajectory of the developer contributions over time. This variable takes the value from one to six for each successive month a developer's contribution is tracked and helps tap the ongoing process of change that affects the rate at which developers' contributions vary over time.

Monthly focal project code commit activity (dependent variable): The dependent variable of interest is the developers' monthly post-survey code commit activity over a period of six months in the focal project (*Focal Project Commits*).[6] The six month observation period is in line with other studies (Joyce and Kraut 2006; Solinger et al. 2013), and allows sufficient data points for a growth-trajectory analysis.[7] We tracked developer activity on GitHub for a six-month period immediately after they responded to the survey. We chose to focus on code commit activity because it is the most direct indicator of the software artifact evolving and is a necessary (but not sufficient) condition for project success (Crowston et al. 2006).

Time-varying covariates (monthly non-focal project activity): Because OSS developers often work simultaneously on multiple projects (Singh and Phelps 2013), their concurrent activities on

---

[6] Focal project is the project they responded to the survey for. If a developer worked on multiple projects that met our sample inclusion criteria, then they were asked about a single project chosen at random to mitigate self-selection bias. Non-Focal project is any other project they were working on in that period but were not asked any specific survey questions for.

[7] However, we also present the results of a robustness check with an alternate time period later.

other (non-focal) projects may influence their contribution to the focal project by diverting their attention away from it (Cummings and Haas 2012; O'Leary et al. 2011). Alternatively, working on other projects may allow developers to tap into efficiencies by balancing the load and applying skills learned from one project to another with greater efficiency (O'Leary et al. 2011). To control for such effects, we tracked their monthly activity on non-focal projects and include the following time-varying covariates in our model: *Non-Focal Project Commits*, *Non-Focal Project Pulls,* and *Non-Focal Project Issues*.

Developer-level covariates: As noted earlier, *Value Fit* focuses on the compatibility derived from the similarity in values, beliefs, and norms, and is thus "supplementary" in nature (Kristoff 1996). However, the person-organization fit literature notes that two other forms of fit, which are "complementary" in nature, also exist when the person or environment provides (or complements) what is missing in the other (Kristof-Brown and Guay 2011; Muchinsky and Monahan 1987). First, "Need-Supply" fit is said to exist when an organization supplies a person's pragmatic needs that are separate from their need for consensual value-based validation (Kristof-Brown et al. 2005). OSS developers also contribute to projects to satisfy various practical needs such as need for software, enhancing their careers, finding enjoyment in programming, feeling competent, peer-recognition, and learning (Shah 2006). Therefore, we operationalize *Need-Supply Fit* as the match between developers' pragmatic needs that they seek to satisfy and the extent to which these are satisfied in a project. Second, "Demand-Ability" fit is said to exist when a person is able to meet the professional demands of an organization (Kristof-Brown et al. 2005). Following Cable and DeRue (2002) we operationalize *Demand-Ability Fit* as the match between a developer's knowledge, skills, and abilities and the demands of the project, which indicates whether a developer has "what it takes" to meet the project's technical demands successfully. Including *Need-Supply Fit* and *Demand-Ability Fit* as developer-level controls in our model allows us to tease out the effect of *Value Fit* and separate it from any effects purely driven by developer abilities and need fulfilment. Appendix-A presents the survey items.

We control for differences in activity levels arising due to developer experience by including the variable *Tenure on GitHub*, which is a count (in years) since the day they registered on GitHub. We include the total *Number of Past Commits in Focal Project* accumulated in the seven-month period prior to the start of our observation window, which indicates developers' sunk costs and their commitment, knowledge, and expertise with the focal project. Core developers have higher sunk costs, as reflected in their past commits, as compared to peripheral developers (Crowston et al. 2006; Setia et al. 2012). Finally, we control for the developers' *Number of Project Associations* and the number of *Followers* of a developer, which is a measure of their popularity and recognition among peers and can be a motivating factor.

Project-level covariates: Newer projects may suffer from the liability of newness and struggle to attract developer attention (Chengalur-Smith et al. 2010). We include the *Project Age* (in years) to control for this effect. Projects with a larger developer base may also suffer from higher developer dropout rates, thus we control for the number of *Developers in Focal Project* (Butler 2001). More active and successful projects may attract a greater number of commits. Thus, we include *Ln (Commits) in Focal Project*, *Releases*, and *Ln (Forks) in Focal Project* (Crowston et al. 2006; Daniel et al. 2013; Setia et al. 2012). Finally, popular projects may attract more contributions, thus we control for *Ln (Stars) in Focal Project* that reflects the popularity rating of the project. Table 2 presents the descriptive statistics of all the variables used in our study.

Table 2: Descriptive Statistics

VARIABLE	**N**	**Mean**	**SD**	**Min**	**Max**	**Source**	**Type**
Developer Monthly Activity Variables (Level 1: Time Varying)							
Focal Project Commits	3384	10.24	25.75	0	324	Archive	Dependent

Month	3384	3.50	1.71	1	6	Archive	Trajectory
Non-Focal Project Commits	3384	48.51	205.40	0	8989	Archive	Control
Non-Focal Project Pull Requests	3384	3.36	8.16	0	106	Archive	Control
Non-Focal Project Issues Reported	3384	2.65	7.05	0	100	Archive	Control
Developer Level Variables (Level 2)							
Tenure on GitHub (Years)	564	3.65	1.56	0.32	6.48	Archive	Control
Number of Past Commits in Focal Project	564	332.35	749.28	3	9929	Archive	Control
Number of Project Associations	564	34.15	33.99	0	175	Archive	Control
Followers (Popularity)	564	64.55	147.68	0	1640	Archive	Control
Paid in Focal Project (Binary)	564	0.35	0.48	0	1	Survey	Predictor
Value Fit	564	0.02	0.98	-2.85	2.15	Survey	Predictor
Need-Supply Fit	564	0	1	-3.64	2.14	Survey	Control
Demand-Ability Fit	564	0.01	1	-4.12	2.07	Survey	Control
Project Level Variables (Level 3)							
Project Age (Years)	431	2.62	1.49	0.11	6.59	Archive	Control
Ln(Commits) in Focal Project	431	7.62	1.69	3.04	13.06	Archive	Control
Releases	431	70.99	344.50	0	6554	Archive	Control
Contributors in Focal Project (Size)	431	140.94	385.75	3	4086	Archive	Control
Ln(Stars) in Focal Project	431	6.88	1.37	1.39	11.17	Archive	Control
Ln(Forks) in Focal Project	431	5.53	1.39	1.61	10.18	Archive	Control

Construct Validity and Principal Component Analysis

The survey items presented in Appendix-A were adapted from Cable and DeRue's (2002) study for the OSS context. We extensively validated the items to ensure maximum content and face validity, construct validity, and reliability of our main construct of interest (*Value Fit*), which is distinct from the covariates *Need-Supply Fit* and *Demand-Ability Fit* (Bahat 2020). Two rounds of Q-sorting were performed (Moore and Benbasat 1991; Petter et al. 2007). In the first round, seventeen MBA students at a large public university in northeastern United States were requested to evaluate and sort the items into three fit constructs. Results of this round demonstrated initial construct validity with overall hit ratios of 86.39% across the three fit constructs. We identified all ambiguous items and modified them accordingly. The refined questionnaire was pilot-tested and refined in the second round based on feedback from fifteen developers belonging to a local chapter of OSS developers to ensure content validity and domain coverage. Finally, we solicited feedback from two academic OSS experts who evaluated the face validity of constructs, phrasing and clarity of items, and adequacy of the domain coverage.

In order to provide further evidence for the construct validity, we conducted principal component analysis (PCA) with Varimax rotation (Hair et al. 2009). Table 3 presents these results. PCA allows us to (1) reduce the dimensionality of our 15 survey items and assess construct validity (convergent and discriminant validity), and (2) calculate the component scores to be used subsequently in the regression analysis (Hair et al. 2009). PCA is most appropriate when used with formative constructs, such as value fit, to create component scores that are linear combinations of the indicators or manifest variables (Darrow and Behrend 2017; Widaman 2007). The Kaiser-Meyer-Olkin measure of sampling adequacy was 0.88, above the commonly recommended value of 0.60, and Bartlett's test of sphericity was significant (χ^2 *(105) = 3225.21, p < .05*). A three-component solution was obtained (eigenvalues of 5.64, 1.71, and 1.44) with the scree-plot showing leveling of eigenvalues beyond that. All the items loaded well onto their respective corresponding components (total variance explained, 57.39%). The diagonal values in the anti-image correlation matrix were over 0.50. The communalities were all above 0.30, confirming that each item shared some common variance with other items, but more importantly, no cross loading was above 0.25. These results provide strong evidence of convergent and discriminant validity (Hair et al. 2009). We retained the component scores for further analysis.

Table 3: Principal Components Analysis

Construct	Item	Value Fit	Need-Supply Fit	Demand-Ability Fit
Value Fit	OSSPractice	0.77		
	Forking	0.73		
	Credit	0.71		
	License	0.71		
	Reciprocate	0.70		
	Sponsorship	0.51		
Need-Supply Fit	Career		0.73	
	Financial		0.70	
	Learning		0.65	

	Recognition		0.58	
	Enjoy		0.54	
	Competent		0.53	
Demand-Ability Fit	Knowledge			0.89
	Skills			0.88
	Abilities			0.83

Notes: Loadings smaller than 0.40 are not reported. Total Variance Explained: 57.39%.
Extraction Method: Principal Components Analysis. Rotation: Varimax with Kaiser Normalization.

Analytic approach

Recall that we tracked developers' post-survey monthly contributions to focal projects over a period of six months. This resulted in the data having a three-level structure: monthly contributions – nested within individual developers – nested within projects.[8] This nested structure of our dataset means that assumptions of independence and homoscedasticity required for linear regression and ANOVA are violated. Moreover, assumptions of compound symmetry and sphericity that are necessary for repeated-measures ANOVA are often violated with longitudinal data. To accommodate these assumption violations, we employed the individual growth modeling approach with three hierarchical levels where we treat developers' monthly time-varying activities as level-1 variates, while developer and project level characteristics are entered at levels 2 and 3, respectively (Raudenbush and Bryk 2002). Because the dependent variable is count, we utilized a three-level Poisson growth model within the HGLM framework.

To obtain precise maximum-likelihood (ML) estimates we rely on the Adaptive Gauss-Hermite Quadrature (AGQ) technique, which outperforms other estimation methods such as the Penalized Quasi Likelihood for count outcomes (Pinheiro and Chao 2006; Raudenbush and Bryk 2002). In contrast to linear mixed models, the likelihood function does not have a closed form for HGLMs, thus requiring numerical integration estimation of random effects to obtain ML

[8] Each developer in our sample is associated with a single focal project, thus a cross-classified design is not required.

estimates.[9] The AGQ method provides unbiased estimates (with an arbitrary degree of accuracy) but at the cost of increasing computational effort (Hartzel et al. 2001; Pinheiro and Chao 2006). The degree of parameter accuracy and computational effort are inversely related to each other and depend on the number of quadrature points (*Q*) chosen for estimation, with higher *Q* values *(~ 5-10)* providing more accurate estimates but at the cost of high algorithm convergence runtimes (Lesaffre and Spiessens 2001). We utilized AGQ estimation with $Q = 15$ points to ensure robust estimates.

Panel A presents our fully specified research model in a three-level hierarchical format for *Focal Project Commits*.[10] All level-1 and level-2 variables (except the dummy, *Paid*) were centered on the group mean, while all level-3 variables were centered on the grand mean to reduce multicollinearity concerns (Raudenbush and Bryk 2002). Checks for multicollinearity indicated no major concerns with the highest VIF statistic being 2.02.

Panel A: Hierarchical Model for Focal Commits

Level-1 (Developer Monthly Activity):

$$Log\ (E\ (Focal\ Project\ Commits|\pi)) = \pi_0 + \pi_1\ (Month) + \pi_2\ (Non\text{-}Focal\ Commits) + \pi_3\ (Non\text{-}Focal\ Pulls) + \pi_4\ (Non\text{-}Focal\ Issues) + e \quad \dots (1)$$

Level-2 (Developer):

$$\pi_0 = \beta_{00} + \beta_{01}\ (Paid) + \beta_{02}\ (Tenure) + \beta_{03}\ (Past\ Commits\ in\ Focal\ Project) + \beta_{04}\ (Number\ of\ Project\ Associations) + \beta_{05}\ (Followers) + \beta_{06}\ (Value\ Fit) + \beta_{07}\ (Need\text{-}Supply\ Fit) + \beta_{08}\ (Demand\text{-}Ability\ Fit) + \beta_{09}\ (Paid\ x\ Value\ Fit) + r_0 \quad \dots (2)$$

$$\pi_1 = \beta_{10} + \beta_{11}\ (Paid) + \beta_{12}\ (Tenure) + \beta_{13}\ (Past\ Commits\ in\ Focal\ Project) + \beta_{14}\ (Number\ of\ Project\ Associations) + \beta_{15}\ (Followers) + \beta_{16}\ (Value\ Fit) + \beta_{17}\ (Need\text{-}Supply\ Fit) + \beta_{18}\ (Demand\text{-}Ability\ Fit) + \beta_{19}\ (Paid\ x\ Value\ Fit) + r_1 \quad \dots (3)$$

$$\pi_q = \beta_{q0} \qquad for\ q = 2 \dots 4; \quad \dots (4)$$

Level-3 (Project):

$$\beta_{00} = \gamma_{000} + \gamma_{001}\ (Project\ Age) + \gamma_{002}\ (Ln\ (Commits)\ in\ Focal\ Project) + \gamma_{003}\ (Releases) + \gamma_{004}\ (Contributors\ in\ Focal\ Project) + \gamma_{005}\ (Ln\ (Stars)\ in\ Focal\ Project) + \gamma_{006}\ (Ln\ (Forks)\ in\ Focal\ Project) + u_{00} \quad \dots (5)$$

$$\beta_{10} = \gamma_{100} + \gamma_{101}\ (Project\ Age) + \gamma_{102}\ (Ln\ (Commits)\ in\ Focal\ Project) + \gamma_{103}\ (Releases) + \gamma_{104}\ (Contributors\ in\ Focal\ Project) + \gamma_{105}\ (Ln\ (Stars)\ in\ Focal\ Project) + \gamma_{106}\ (Ln\ (Forks)\ in\ Focal\ Project)$$

[9] Modeling random effects for developers also accounts for over-dispersion in the discrete repeated observations modeled with the Poisson model (Hartzel, Agresti, and Caffo, 2001).

[10] We also present the "mixed" version of the same model in Panel B (in Appendix B) for readers who prefer the linearized format.

… (6)

$\beta_{0m} = \gamma_{0m0}$ *for m = 1…9* … (7)

$\beta_{1n} = \gamma_{1n0}$ *for n = 1…9* … (8)

$\beta_{p0} = \gamma_{p00}$ *for n = 2…4* … (9)

Note: All Level-1 and Level-2 variables (except Paid) were centered on the group mean. All Level-3 variables were centered on the grand mean.

The model in Panel A allows us to analyze the effect of developer and project level predictors on both the code contribution levels at a specific point in time and its rate of change over time, i.e., growth (Willett 1997). Equation 1 represents developers' unique growth trajectories. Due to centering, the intercept for code contribution, π_0, refers to developers' code contribution levels at $t=3$ months, halfway through the observation period; while the slope π_1 refers to the (linear) growth in code contributions over the six-month observation period. The individual size and growth parameters (π_0 and π_1) in equation 1 subsequently become dependent variables that are predicted using developer-specific (the βs in equations 2-4) and project-specific factors (the γs in equations 5-9). Thus, the level-2 and level-3 models predict individual growth trajectories using developer and project characteristics respectively (Raudenbush and Bryk 2002). The most important feature of our model is that it captures the variability in developers' code contributions levels and growth simultaneously. This distinction is important to consider because individual developers differ markedly in their contribution patterns. While some developers have high contribution levels at any given observation time point but slower growth over time, others may have a lower contribution levels but exhibit faster growth over time as they gain experience (Willett 1997).

Results

We began by specifying the null model and then incrementally estimating three conditional models that improved model fit, with the fourth model representing the full model (Panel A). These four models (Models 0-3) are summarized in Table 4 and described below.

Table 4: Results

Variable Type	Fixed Effects	Model 0 Null	Model 1 Covariates	Model 2 Main Effects	Model 3 Interaction
Mean Code Commit Level	Intercept (π_0)	3.40*** (0.01)	3.21*** (0.02)	2.87*** (0.03)	3.28*** (0.03)
Mean Growth Rate	Month (π_1)		0.15*** (0.00)	0.12*** (0.01)	0.33*** (0.01)
Time Varying Covariates					
Level 1 Control	Non-Focal Commits (π_2)		-0.00*** (0.00)	-0.00*** (0.00)	-0.00*** (0.00)
	Non-Focal Pulls (π_3)		0.01*** (0.00)	0.01*** (0.00)	0.01*** (0.00)
	Non-Focal Issues (π_4)		0.00*** (0.00)	0.00*** (0.00)	0.00*** (0.00)
Code Commit Level Covariates (at t=3 months)					
Developer Control (Level 2)	Tenure (β_{02})		0.17*** (0.02)	0.11*** (0.02)	0.09*** (0.02)
	Past Commits in Focal Project (β_{03})		0.00*** (0.00)	0.00*** (0.00)	0.00*** (0.00)
	Project Associations (β_{04})		-0.00** (0.00)	-0.00 (0.00)	-0.00* (0.00)
	Followers (β_{05})		0.00*** (0.00)	0.00 (0.01)	0.00 (0.00)
	Need-Supply Fit (β_{07})		0.44*** (0.03)	0.36*** (0.03)	0.36*** (0.02)
	Demand-Ability Fit (β_{08})		0.41*** (0.02)	0.42*** (0.02)	0.43*** (0.01)
Project Control (Level 3)	Project Age (γ_{001})		0.05** (0.01)	0.06*** (0.02)	0.07*** (0.02)
	LnCommits (γ_{002})		0.25*** (0.01)	0.21*** (0.01)	0.21*** (0.01)
	Releases (γ_{003})		-0.00 (0.00)	-0.00** (0.00)	-0.00* (0.00)
	Contributors (Size) (γ_{004})		-0.00*** (0.00)	-0.08*** (0.00)	-0.00*** (0.00)
	LnStars (γ_{005})		0.16*** (0.02)	0.19*** (0.02)	0.21*** (0.02)
	LnForks (γ_{006})		-0.09*** (0.03)	-0.11*** (0.03)	-0.12** (0.03)
Growth Trajectory Covariates (Change Over Time)					
Developer Control (Level 2)	Tenure (β_{12})		0.08*** (0.00)	0.07*** (0.00)	0.06*** (0.00)
	Past Commits in Focal Project (β_{13})		-0.00* (0.00)	-0.00** (0.00)	-0.00** (0.00)
	Project Associations (β_{14})		0.00*** (0.00)	0.00*** (0.00)	0.00*** (0.00)
	Followers (β_{15})		-0.00*** (0.00)	-0.00*** (0.00)	-0.00*** (0.00)
	Need-Supply Fit (β_{17})		0.01 (0.01)	0.01** (0.00)	0.02*** (0.00)
	Demand-Ability Fit (β_{18})		0.00 (0.01)	0.00 (0.00)	0.00 (0.00)
Project Control (Level 3)	Project Age (γ_{101})		0.02*** (0.00)	0.02** (0.00)	0.02*** (0.00)
	LnCommits (γ_{102})		0.01** (0.00)	0.00 (0.00)	0.00 (0.00)
	Releases (γ_{103})		-0.00*** (0.00)	-0.00*** (0.00)	-0.00*** (0.00)
	Contributors (Size) (γ_{104})		-0.00*** (0.00)	-0.00*** (0.00)	-0.00*** (0.00)
	LnStars (γ_{105})		0.01 (0.01)	0.01 (0.01)	0.00 (0.01)
	LnForks (γ_{106})		-0.01 (0.01)	-0.01 (0.01)	-0.00 (0.01)
Code Commit Level Predictors (at t=3 months)					

Predictor of Developer Code Commit Level	Paid (β_{01})			0.69*** (0.05)	0.76*** (0.08)
	Value Fit (β_{06})			0.11*** (0.02)	0.08*** (0.01)
	Paid X Value Fit (β_{09})				0.10** (0.04)
Growth Trajectory Predictors (Change Over Time)					
Predictor of Developer Growth Trajectory	Paid (β_{11})			0.04*** (0.01)	0.04*** (0.01)
	Value Fit (β_{16})			0.05*** (0.01)	0.11*** (0.00)
	Paid X Value Fit (β_{19})				-0.13*** (0.01)
Model Fit Statistics					
Deviance (-2 log likelihood)		34638.01	27772.20	27802.60	27541.20
Estimated Parameters		3	33	37	39
AIC		34644.01	27838.20	27876.60	27619.27
BIC		34648.55	27888.13	27932.57	27678.27

Notes: * 0.10; ** 0.05; *** 0.01; Robust Standard Errors are in parentheses; Poisson Full Maximum Likelihood via Adaptive Gaussian Quadrature (*Q=15*)

Model 0: The unconditional (or null) model has no predictors at any level and allows partitioning the variance across the developer and project levels (Raudenbush and Bryk 2002). If the variance attributable to the project level is not significant then the model can be simplified to a two-level design. Results show that the variance due to developer characteristics was $r_0 = 4.38$; and the variance at the project level was $u_{00} = 1.13$, which was significantly different from zero ($\chi^2 = 559.02$; $df = 418$; $p < 0.01$), thus necessitating a three-level design. The intra-class coefficient (ICC), which measures the proportion of variance at the project level (level 3), shows that a significant proportion of variance in monthly commit activity is due to project level characteristics ($1.13 / (4.38+1.16) = 20.39\%$). The first column in Table 4 presents the results of the null model. The intercept is significant ($\pi_0 = 3.40$, $p < .01$), and represents the natural log of the expected (average) code commit activity across the population (*29.96*) at $t=3$ months.[11]

Model 1: We then estimated a covariates-only model. Here, the time-varying covariates *Month* and monthly contributions to non-focal projects (*Non-Focal Project Commits*, *Non-Focal Project Pulls,* and *Non-Focal Project Issues*) are entered at level-1 (equation (1) in Panel A).

[11] HGLM uses the log-link function for level-1 Poisson models, i.e., the natural log of the event rate (Raudenbush and Bryk, 2002).

These intra-individual factors capture the temporal dependence of developers' code contributions and growth over time (Willett 1997). The growth parameter, π_1, is positive and significant (π_1 = 0.15, p < .01), suggesting overall upward contribution trajectories. The model also includes other developer-level covariates (*Need-Supply Fit*, *Demand-Ability Fit, Tenure, Past Commits in Focal Project, Number of Project Associations, and Followers*), and project-level covariates (*Project Age, Ln (Commits) in Focal Project, Releases, Developers in Focal Project, Ln (Stars) in Focal Project, and Ln (Forks) in Focal Project*), which are entered at levels 2 and 3 respectively as controls for the code commit levels (equations (2) and (5) in Panel A) and the growth in code commits (equations (3) and (6) in Panel A).

Model 2: In this model we introduce the main effects of *Value Fit* and *Paid* to predict the parameters for both code commit level (π_0 in equations (2) and (5) in Panel A), and growth (π_1 in equations (3) and (6) in Panel A).

Model 3: Finally, we augment the main effects (model 2) by incorporating the interaction between *Value Fit* and *Paid* to predict both code commit level (equations (2) and (5)) and growth (equations (3) and (6)).

We compared these nested models for improvement in model fit. In HGLM, the deviance difference between two models (one nested in the other) is chi-squared distributed, with a degree of freedom equal to the difference in number of estimated parameters (Raudenbush and Bryk 2002). The deviance also allows the calculation of information theoretic model selection criteria (e.g., AIC, BIC) that reward high fit but penalize unnecessary complexity. The deviance difference test reports that model 3 is significantly better fit than the main-effects model 2 ($\Delta\chi^2_{(2-3)} = 261.40, df = 2, p < .01$) and the covariates-only model 1 ($\Delta\chi^2_{(1-3)} = 231.00, df = 6, p < .01$). Thus, incorporating the interaction effects significantly improved model fit, and helps explain variance beyond models that include only the main effects and controls. Furthermore, both AIC and BIC achieve minima for model 3, suggesting that it is the best model among the cohort. Therefore, we focus on model 3 estimates to test our hypotheses (Table 4).

The main effects of *Value Fit* are positive and statistically significant for both the code contribution level at $t=3$ months ($\beta_{06} = 0.08, p < 0.01$) and growth in code contributions ($\beta_{16} = 0.11, p < 0.01$). These results provide strong evidence for H1a and H1b. In addition, although not hypothesized, the main effect of *Paid* is positive and significant for both the code contribution level at $t=3$ months ($\beta_{01} = 0.76, p < 0.01$) and growth ($\beta_{11} = 0.04, p < 0.01$). Taken together, these results point to the overall positive effect of value fit (and payment) on not only developers' code contributions but also on the growth over time.

To test for H2, we analyzed the interaction effect *Paid x Value Fit* which was significant and positive for code contribution level at $t=3$ months ($\beta_{09} = 0.10, p < 0.05$). The positive sign on the β_{09} interaction coefficient suggests that the effect of value fit on code contribution level is stronger for paid compared to unpaid developers. In contrast, the effect of *Paid x Value Fit* was significant but negative for the change over time, i.e., for the growth in code contribution ($\beta_{19} = -0.13, p < 0.01$). The negative sign on β_{19} coefficient suggests that the effect of value fit on the growth in code contributions over time is stronger for unpaid than paid developers. These results provide evidence to support both H2a and H2b. Figure 2 illustrates the main and interaction effects. As shown, the trajectories for developers reporting stronger value fit are higher than developers with weaker value fit. This is apparent for both paid and unpaid developers; however, paid developers have higher number of commits than unpaid developers (assuming the same level of value fit) on average. The trajectories for both paid and unpaid developers have upward slopes initially, however, paid developers reach an inflection point (between months four and five) after which their growth plateaus and actually exhibits downward slope, while unpaid developers see a much more robust and continuous growth throughout the six-month observation window. These observations are in line with our hypotheses.

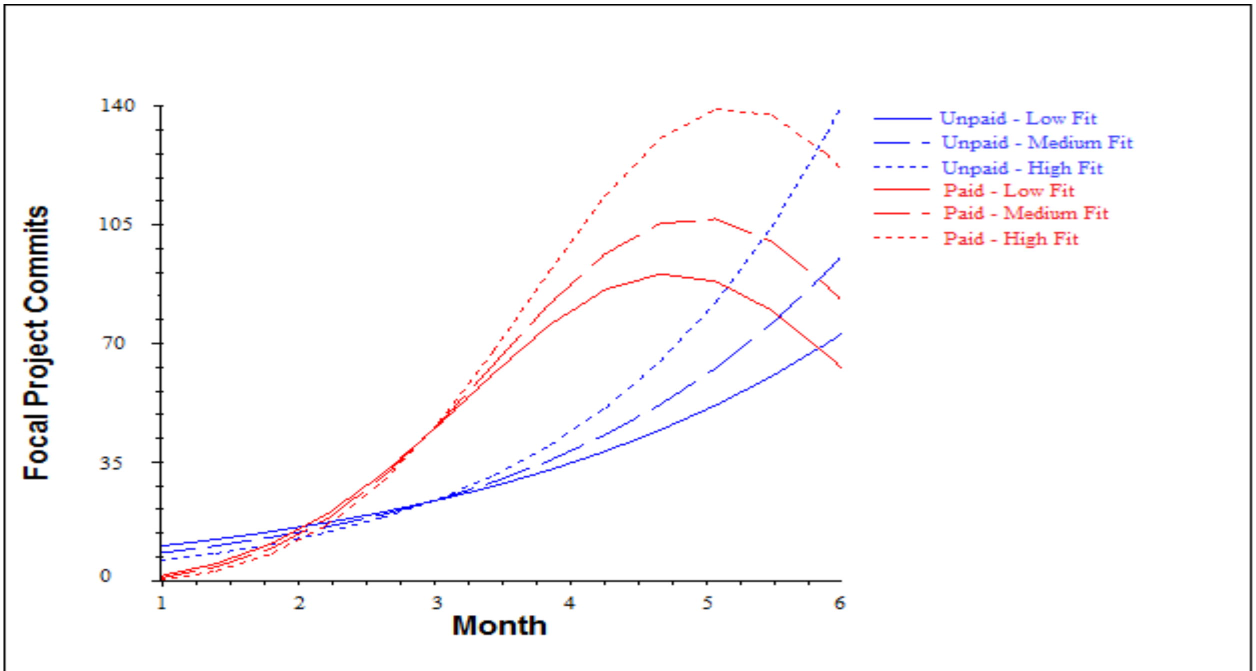

Figure 2. Effects of Value Fit and Payment over Time

Robustness Checks

To explore how generally applicable our findings are, we utilized an alternative observation period to check the robustness of results. In the analyses thus far, we have utilized an observation period of six months, which coincides with other studies and allows sufficient data points (e.g., Joyce and Kraut 2006). On the other hand, significantly shorter time periods can reduce the statistical power and variance available for a longitudinal growth analysis, and a minimum of three observation periods are required (Raudenbush and Bryk 2002). A counterargument to the six-month observation period could be that the developers' situation may evolve during this extended timeframe, thus affecting the trajectory. To check for the robustness of our results, we selected an alternative (shorter) period of four months to run our model. These results (Appendix-C) provide support for our results with one exception. While the main effect of *Value Fit* was a significant predictor of the code contribution at $t=3$ months (six-month model), it was not significant at $t=2$ months in the four-month model. This difference could be attributed to the reduced power due to the shorter observation period.

Next, we note that HGLM provides two types of estimates: unit-specific (US) and population-average (PA). PA estimates allow testing the differences in average response between two groups with different risk factors. The PA coefficients have interpretation for the entire population rather than any specific individual. All results presented thus far utilized the population-average estimates with robust standard errors for two reasons: (1) PA responses are more robust to misspecifications in the random effects, and (2) our goal is to study how the code contribution and its growth differs between paid and unpaid developers (with high or low fit) across all projects in the population. Thus, we analyzed the differences in contribution rates associated with an increase in predictors, holding constant the other predictors, but averaged over all project-level random effects. On the other hand, US estimates explicitly take into account any subject-specific heterogeneity and represent an individual developer's response to the risk factors (e.g., being paid). We also present US estimates (Appendix-C) and note that they agree with the population-average estimates. Finally, we also tested a quadratic growth model that takes into account the acceleration in growth rate by including the quadratic term, $Month^2$, in our model (Appendix-C). These results are also consistent with our main results.

Discussion

OSS development framework finds itself at the crossroads where motivations based on shared values, norms, and beliefs (i.e., value fit) collide with the market dynamics and financial incentives (Berdou 2011). This is reflected in more than a decade-long, continuing debate in the

Debian community regarding whether to allow, and if so, how to align payment with developers' motivations.[12, 13] Thus, how payment interacts with motivational factors that have been the dominant drivers of OSS participation is a critical, yet unresolved question. Using the motivation-hygiene framework, we posed the following question: *How does the interaction between developers' perceptions of value fit with an OSS project and payment impact their code contribution levels and its rate of change over time (i.e., growth)?* Table 5 summarizes our hypotheses and findings. We discuss the contributions to research and implications next.

Table 5: Summary of Findings

Hypothesis	Supported
H1a: Value fit with the project positively affects developer code contribution.	Yes ($\beta_{06} = .08; p < 0.01$)
H1b: Value fit with the project positively affects the growth in developer code contribution over time.	Yes ($\beta_{16} = .11; p < 0.01$)
H2a: The positive effect of value fit on code contribution is stronger for paid than unpaid developers.	Yes ($\beta_{09} = .10; p < 0.05$)
H2b: The positive effect of value fit on the growth in code contribution over time is stronger for unpaid than paid developers.	Yes ($\beta_{19} = -.13; p < 0.01$)

Contributions to Research

Past OSS research has identified factors that motivate developers to contribute (e.g., Hars and Ou 2002; Shah 2006; Von Krogh et al. 2012), noted that developers can be motivated by several factors simultaneously (e.g., Hann et al. 2013; Lakhani and Wolf 2005; Roberts et al. 2006; Shah 2006), and identified the importance of long term contributions for project survival (e.g., Chengalur-Smith et al. 2010; Fang and Neufeld 2009). Despite this strong foundation, the OSS literature has yet to point out why developers show distinct code contribution patterns over time (Lee 2018; Wang et al. 2018). Further, the focus on perceptive outcome measures such as intentions to continue (e.g., Ghosh et al. 2013; Schilling et al. 2012; Ho and Rai 2017), or on

[12] https://lwn.net/Articles/201488/
[13] https://lwn.net/Articles/790954/

number of code commits (e.g., Maruping et al. 2019) ignores the growth in contributions, which offers a more holistic assessment of how developers' actual coding behaviors will evolve in the future. By focusing both on the level of code contribution and growth in code contributions over time, we contribute to the OSS motivation literature in several ways.

First, our finding that value fit increases the rate of code contribution over time is consistent with Von Krogh et al.'s (2012) strong emphasis on understanding OSS development as a social practice. They argue that a sense of community drives developers to contribute. Our conceptualization of value fit, which aligns a developer with the project environment based on shared values, as a strong antecedent of long-term contributions underscores the continued importance of such motivational factors, even in the presence of financial incentives and corporate involvement in the evolving OSS framework. Developers who perceive value fit contribute more regardless of pay. Value fit matters – it amplifies contributions over time.

In contrast, differences in hygiene factors (paid or not) lead to fundamentally distinct contribution patterns. Unpaid developers who perceive high value fit exhibit an increase in contribution levels over time, while paid developers show higher levels of contributions initially but decreasing growth in the long-term. A sharper focus on time explains when pay has a positive or negative impact on code contribution. Highlighting the critical role played by time also helps us synthesize prior findings that seemed to contradict each other. In particular, we help reconcile the findings in OSS developer motivation literature with the prior findings in psychology literature regarding the tension between intrinsic motivations, which are analogous to Herzberg's motivation factors, and extrinsic motivations, which are analogous to Herzberg's hygiene factors (Sachau 2007). Scholars in psychology have long suggested that extrinsic rewards crowd out intrinsic rewards (Kohn 1999). The research on the negative effects of external rewards began with Deci (1971), and since then numerous meta-analyses studies (e.g., Deci et al. 1999; Wiersma 1992) have provided strong empirical evidence that tangible financial rewards (e.g., payment) for engaging, performing, or completing enjoyable and creative tasks reduce interest in the activity (Sachau 2007). Despite these strong findings, recent research in the OSS context has disputed this account (Alexy and Leitner 2011; Hars and Ou 2002; Lakhani and Wolf 2005; Roberts et al. 2006). It is notable that this stream of OSS research did not consider developers' code contribution trajectories. Our results show that the two divergent narratives can be reconciled when considering time as an important ingredient in understanding OSS contributions, and in particular, by considering the distinction between the levels of code contribution versus its growth over time, which represent two separate, yet equally important aspects of code contribution trajectories (Raudenbush and Bryk 2002). We find that motivation factors like value fit, and hygiene factors like payment, can complement each other in the short-term resulting in higher code contribution levels, thus aligning our findings with the OSS stream. However, receiving payment inhibits the positive effect of value fit on the growth in code contribution, bringing our results more in line with the psychology literature, which argues that payment can dampen motivations over time. Unpaid developers – who have freedom to choose interesting tasks – showed stronger positive effect of value fit on the growth of code contributions than paid developers who are obligated to work. This makes sense because hygiene factors typically produce only short-term highs, but their motivating effects dissipate over the long-term as the paid individual gets habituated to the current level of remuneration, which becomes their minimum level expectation for the future (Brickman et al. 1978; Sachau 2007). This phenomenon has been well documented in the literature on materialism where it has been variously called "hedonic adaptation" (Frederick and Lowenstein 1999), "rising baselines"

(Kasser 2002), “hedonic treadmill” (Brickman and Campbell 1971) etc.

Second, the OSS developer motivation literature has focused on either the person or project (environment) level factors separately but not on the *match* between them. By conceptualizing both person and environment dimensions simultaneously in this context, we accommodate developers’ perceptions of the match between their personal and project values which has been shown to be a much stronger predictor of outcomes than either the individual or environment characteristics alone (Kristof-Brown et al. 2005; Kristof 1996). Our main construct, value fit, represents a match in the value, norms, and beliefs of the developer and the project community. Because such values are durable and persist over time (Oreg and Nov 2008; Stewart and Gosain 2006; van Vianen et al. 2007), value fit – once achieved – is likely to be relatively stable. However, developers may evolve more readily on the complementary fit dimensions, especially the Demand-Ability fit, which is expected to evolve as the developer’s experience, skill, and life-situation changes over time, along with the project’s technical and effort demands as it progresses. This represents a fertile opportunity for future investigation.

Third, OSS research has generally focused on identifying developer motivations (Hars and Ou 2002), but not necessarily on how to manage their motivations to facilitate long-term contribution. Our work takes the first step toward moving the attention to the “job enrichment” perspective, which relates to how employers can create conditions that keep employees motivated (Sachau 2007). In particular, our work opens up research opportunities in how to best manage the precious resource of unpaid developers because volunteer management remains an overlooked aspect not only in the management research in general (Ertas 2019), but in the OSS domain in particular. This understanding is important for employers and project leaders to strategically facilitate long-term OSS developer code contributions. Unpaid developers need to feel that their work is being valued and respected. While unpaid developers who do not feel they fit well with the project's values have the option to simply leave, contractually bound paid developers who do not fit may find their motivation wane, thereby adversely affecting their performance. Conversely, developers who perceive a strong sense of value fit will be more resilient, even when they are not paid for their contributions. How can OSS projects best articulate their values and communicate them efficiently to attract skilled volunteer developers who share the same value system? What kind of recruiting processes can organizations use to leverage value fit when staffing paid developers to work on OSS projects? These are worthwhile avenues of investigation. Further, we hypothesized that payment results in slower growth in code contributions because its positive effects dissipate over time as developers get accustomed to the level of pay, and paid developers feel inhibited by contractual constraints that reign in their motivation. Short of providing a continual increase in pay that may not always be possible or even desirable, how can this be avoided? The volunteer management literature has discussed the role of paid staff as "quasi volunteers" in NGOs to motivate them better, i.e., employing them as paid staff who also have some of the creative freedoms enjoyed by volunteers (Kreutzer and Jäger 2011). Can allowing paid OSS developers more creative freedom (e.g., in choosing interesting tasks, not just routine maintenance) and relieving some of the contractual constraints help alleviate their decrease in growth in contributions (Figure 2)? This question opens up research avenues regarding how to best align paid developers' roles vis-à-vis unpaid developers.

Finally, research shows that when volunteers and paid staff work together in a volunteer-involving organization (e.g., nonprofits), they often perceive the organization's values through distinct personal lenses leading to disagreements (Kreutzer and Jäger 2011). When unpaid developers perform the same coding tasks as paid developers equally well, it is natural for paid developers to feel threatened while unpaid developers to feel underappreciated. The Debian project highlights that similar issues arise in the OSS context as well (Gerlach et al. 2016), and that management processes need to be developed to mitigate such issues. NGOs often employ volunteer resource managers who help oversee issues in maintaining relationship between paid staff and volunteers (Ertas 2019). Can certain individuals act as liaison between volunteer and paid OSS developers to manage expectations and interactions better? These are worthwhile questions, and more research is needed to develop practices that best align OSS motivations and financial interests.

Implications for Practice

OSS communities face increasing challenges in organizing developers influenced by psychological and economic motivations (Gerlach 2016). As the variety of motivations increase, managers may face difficulty in understanding what to expect from developers over time. Corporate, government and educational sectors have begun to pay many OSS developers, and managers and project leaders alike must understand how payment interacts with motivation factors like value fit to influence sustained developer contributions. Given that code commits are central to development, our results point to the limits in the efficacy of financial incentives in eliciting code contributions in the long-term. Corporate sponsors should not rely on pay as the exclusive mechanism and overlook the impact of motivators like value fit. It might be tempting for managers and firms to focus solely on financial rewards and ignore the hard work that goes in creating an interesting environment for work. Our results reveal the limits of such a strategy in the OSS context, especially when the managerial interest is focused on the growth in contributions over time. Indeed, the Linux foundation (2019) also notes that while payment can get OSS developers in the door, retaining them requires strategies focused on creating an environment conducive to their perceptions of value fit.

The OSS literature and practice have thus far treated developer motivations as a given. The focus has been on identifying what motivates volunteer developers to contribute, with the implicit assumption that self-motivated developers will contribute to OSS with minimal guidance or encouragement in self-organizing teams (Crowston et al. 2007). However, due to the new hybrid OSS work environment, we advocate the introduction of the "job enrichment" perspective and the associated human resource practices, which can be particularly helpful in enhancing the motivations of paid OSS developers by giving desired responsibilities and freedom to choose interesting work and combat the limitations of hygiene factors (Sachau 2007). Hiring paid developers to simply do mundane, uninteresting tasks while ignoring their motivational factors may not be a worthwhile strategy in the long-term. The first step toward this goal is for the managers and firms to recognize that contractual obligations can limit paid developer's ability for psychological growth and rein in their motivation. Furthermore, paid developers are companies' de facto brand ambassadors who can ensure continuity in project development. Yet, they can be sensitive about balancing the corporate and community interests and tend to avoid tasks that go against community values (Berdou 2011). Companies should invest in helping them walk the precarious tightrope by reducing constraints that bind them and provide proper training. Companies should also consider offering them more autonomy in selecting their tasks and chances to learn new skills. We acknowledge that relinquishing control is easier said than done when competing interests are at stake. OSS project leaders can potentially help increase paid developer contribution by promoting relationships between these developers and the community. In many ways, paid OSS developers are unique due to the hybrid nature of their work environments and the peers they collaborate with. They may require specialized human resource-based practices to motivate and help them manage relationships with volunteer developers.

From the point of view of managing the volunteer OSS workforce, senior members and peers can be highly influential agents in helping create an environment conducive to developer motivations and value fit (Moreland and Levine 1982). Value fit can be a crucial determinant of developer retention, in particular when other factors such as learning and skill development plateau over time (Zhou et al. 2016). Furthermore, prior research suggests that developers' propensities to contribute long-term may differ depending on their personality types. For example, Furnham et al. (1999) found that extraverts put more emphasis on motivational factors, so long as hygiene factors are not problematic, when they choose a job, suggesting that managerial strategies may be more effective in eliciting their long term contributions by helping reinforce their sense of value fit with the community. On the other hand, introverts are driven more by hygiene factors and may avoid contributing to projects without pay. It may be useful for project managers to be mindful of differences in developers' personality traits so that they are managed according to the factors they value more.

Limitations

We acknowledge some limitations in our study and offer avenues for future research. By building a bridge between the motivation-hygiene and fit literatures our work opens new avenues of investigation into how different types of fit may influence contributors in online peer production environments. While we find that value fit is an important factor in determining long term contributions, there other aspects of fit that are also worthy of exploration that were not in the scope of our study. For example, the person-environment fit literature distinguishes between person-job, person-supervisor, and person-group fit as distinct concepts to assess individual-level outcomes such as attraction, joining, retention, and withdrawal (Kristoff-Brown et al. 2005). This can be a fruitful avenue to explore because OSS projects follow different types of governance models, which may make subtle differences in the different aspects of fit salient (O'Mahony and Ferraro 2007). For example, while many OSS projects are supervised by a so-called benevolent dictator (e.g., Linux), others follow a rotating dictatorship model (e.g., Perl), or even a democratic voting-based approach as followed by Apache (Ljungberg 2000). Additionally, both core and peripheral OSS developers perform a wide variety of tasks in projects requiring different skills (Setia et al. 2012). Our model controls for the total number of past commits in the focal project accumulated in the seven-month period prior to the start of our observation window, which indicates not only a developer's sunk cost but also their commitment, knowledge, and expertise with the focal project. Core developers are likely to have much larger sunk costs, as reflected in the number of past commits in the focal project, than peripheral developers (Crowston et al. 2006; Setia et al. 2012). Yet, it remains unclear how core and peripheral OSS developers' perceptions of fit with the type of jobs they have, tasks they perform in projects, their backgrounds, their supervisor(s), the project's governance model, or fit with their employers' OSS policies impact outcomes. In a related vein, it would be worthwhile to explore whether it is the fit with a specific project or group, or fit with the broader OSS community that matters more when determining their continued participation to OSS, and how do these forms of fit interact with each other.

The use of the dichotomous variable for payment does not allow us to assess the impact of the magnitude or type of pay. Furthermore, it does not allow us to assess whether an employer requests the stop and start of code contributions at various times. Several OSS developers who helped us pilot test the survey items advised us during informal conversations that detailed questions about payment (i.e., amount or source) may be considered too sensitive, and respondents would be reluctant to provide details. We leave it to future research to explore whether the type of employer (e.g., for-profit, non-profit etc.) or the amount of pay alters the interaction between motivation and hygiene factors. Delving into funding sources such as *patreon* and *opencollective* represents another area worthy of consideration. Finally, the effect of other types of financial rewards such as bonuses, benefits, and promotion should be considered.

We focused on the quantity of code contribution as our dependent variable because code commits are the primary indicator of the software artifact evolving and indicate a necessary (but not sufficient) condition for project success (Crowston et al. 2006). In addition, code contributions offer more opportunities for psychological fulfilment and lead to increased responsibility and rank within the project compared to other tasks like documentation. Despite the benefits of this measure, limitations remain that offer opportunities for future research. Code quality is also an important factor to consider because it influences user interest and the ease with which the application can be modified in the future (MacCormack et al. 2006). Furthermore, OSS projects also depend on other types of important contributions including bug reports, feature requests, documentation, and coordination among developers and stakeholders. Future research should assess the generalizability of our results by assessing how motivation and hygiene factors affect such contributions. It is also worth considering that after making code contributions developers may transition into management roles and therefore make fewer commits (Dahlander and O'Mahony 2011). The relationship between motivations and work may also change over the course of a developer's professional's lives (DeLone and McLean 1992), suggesting that stage of career may be another important boundary condition for the impact of motivation on outcomes. In this vein, it might be useful to assess how Need-Supply and Demand-Ability fit evolve, as developers gain in experience to improve their skills and abilities, and their needs change over time. Another opportunity to extend our model rests in the addition of satisfaction and dissatisfaction as mediators of the impact of value fit and pay on code contribution, which were not included in this study. We expect value fit to positively impact satisfaction, and satisfaction to lead to more contribution over time, while pay (or lack thereof) should lead to dissatisfaction and lower code contribution over time.

Our sample was limited to a survey of developers from a single platform, GitHub. Although GitHub is the largest OSS repository (Lindberg et al. 2016), it does not host the largest OSS projects (e.g., Linux). Larger projects like Linux may be more likely to attract paid developers, and as such replicating our study on large OSS projects presents an opportunity for future research. In addition to the size, the provenance of OSS projects may also influence the developers' perceptions of fit and code contributions to the project (Ho and Rai 2017). Our sample was composed predominantly of male developers. While most other researchers have found OSS communities to be predominantly male (Bagozzi and Dholakia 2006; Choi and Pruett 2015), this brings up the question about the degree to which our model would fit a community that has different gender distribution. Finally, because we restricted our attention to a six-month period to assess developer trajectories, we are unable to assess how developer trajectories evolve beyond this point. While we controlled for the acceleration or deceleration in developer trajectories via the inclusion of the quadratic term (Appendix C), we leave the exact nature of the trajectory shape for future research. Given the human time constraints, the growth of volunteer code contributions shown in figure 2 is not sustainable forever. In addition, paid developers might eventually settle down to linear, more horizontal trajectories after exhibiting the downward slope for certain period of time, in which case the trajectory may not have an exact curvilinear (or inverted-U) shape. Future research should explore how long this growth persists until it begins to plateau by observing developers for a longer period of time.

Conclusion

OSS production is increasingly taking place in hybrid communities where both hygiene and motivation factors influence developer contributions. The increasing prevalence of financial incentives for OSS development raises important issues regarding developers' contributions in the short and long run. Utilizing Herzberg's motivation-hygiene theory, this study untangles the complex effects of motivation versus hygiene factors on developer contribution both cross-sectionally and longitudinally. Our findings illustrate long term positive effects of value fit on developer contribution to OSS projects and its growth, highlighting its importance even in the presence of financial incentives and corporate engagement. We also document important differences in the effect of value fit between paid and unpaid developers with regard to their level of code contribution and change over time.

References

Alexy, O. and Leitner, M. 2011. "A fistful of dollars: Are financial rewards a suitable management practice for distributed models of innovation?," *European Management Review* (8:3), pp. 165-185.

Atiq, A. and Tripathi, A. 2014. "Monetary Rewards for Open Source Software Developers*," Thirty Fifth International Conference on Information Systems*, Auckland, pp.

Bahat, E. 2020. Person–Organization Fit and Commitment to Volunteer Organizations. *VOLUNTAS: International Journal of Voluntary and Nonprofit Organizations*, pp. 1-16.

Bagozzi, R. and Dholakia, U. 2006. "Open Source Software User Communities: A Study of Participation in Linux User Groups," *Management Science* (52:7), pp. 1099-1115.

Barcomb, A., Kaufmann, A., Riehle, D., Stol, K.-J. and Fitzgerald, B. 2018. "Uncovering the Periphery: A Qualitative Survey of Episodic Volunteering in Free/Libre and Open Source Software Communities," *IEEE Transactions on Software Engineering*, pp. 1-18.

Berdou, E. 2011. Organization in Open Source Communities: At the Crossroads of the Gift and Market Economies. Routledge.

Bergquist, M., Rolandsson, B. and Ljungberg, J. 2009. "Open source programmers strategies to cope with ideological tensions*," International Conference on Organizational Learning, Knowledge and Capabilities*, Amsterdam, the Netherlands.

Boutyline, A. and Willer, R.. 2017. "The social structure of political echo chambers: Variation in ideological homophily in online networks," *Political Psychology* (38:3), pp. 551-569.

Brickman, P., and Campbell, D.T. 1971. *Hedonic relativism and planning the good society*. In M.H. Apley (Ed.), Adaptation-level theory: A symposium (pp. 287-301). New York: Academic Press.

Brickman, P., Coates, D., and Janoff-Bulman, R. 1978. "Lottery winners and accident victims: Is happiness relative?" *Journal of personality and social psychology*, (36:8), pp. 917-927.

Brock, A. 2013. "*Understanding Commercial Agreements with Open Source Companies,"* Brussels: OpenForum Europe LTD for OpenForum Academy.
Brodkin, J. 2014. "Tech giants, chastened by Heartbleed, finally agree to fund OpenSSL," available at: https://arstechnica.com/information-technology/2014/04/tech-giants-chastened-by-heartbleed-finally-agree-to-fund-openssl/; accessed June 13th 2019.
Butler, B. 2001. "Membership Size, Communication Activity, and Sustainability: A Resource-Based Model of Online Social Structures," *Information Systems Research* (12:4), pp. 346-362.
Byrne, D. 1961. "Interpersonal Attraction and Attitude Simlarity," *Journal of Abnormal and Social Psychology* (62:3), pp. 713-715.
Cable, D.M. and DeRue, D.S. 2002. "The convergent and discriminant validity of subjective fit perceptions," *Journal of applied psychology* (87:5), pp. 875.
Chengalur-Smith, I., Sidorova, A. and Daniel, S. 2010. "Sustainability of Free/Libre Open Source Projects: A Longitudinal Study," *Journal of the Association for Information Systems* (11:11), pp. 657-683.
Choi, N. and Pruett, J.A. 2015. "The characteristics and motivations of library open source software developers: An empirical study," *Library & Information Science Research* (37:2), pp. 109-117.
Corbet, J., Kroah-Hartman, G. and McPherson, A. 2012. "Linux kernel development. How Fast it is Going, Who is Doing It, What They are Doing, and Who is Sponsoring It," *The Linux Foundation*, 1-13.
Crowston, K., Howison, J. and Annabi, H. 2006. "Information systems success in free and open source software development: theory and measures," *Software Process: Improvement and Practice* (11:2), pp. 1223-1148.
Crowston, K., Wei, K., Howison, J. and Wiggins, A. 2012. "Free/Libre open-source software development: What we know and what we do not know," *ACM Computing Surveys* (44:2), pp. 1-35.
Crowston, K., Li, Q., Wei, K., Eseryel, U.Y. and Howison, J. 2007. " Self-organization of teams for free/libre open source software development*," Information and Software Technology* (49:6), pp. 564-575.
Crowston, K., Wei, K., Li, Q., & Howison, J. 2006. "Core and periphery in free/libre and open source software team communications," In Proceedings of the *39th Annual Hawaii International Conference on System Sciences*, 6, IEEE.
Cummings, J.N. and Haas, M.R. 2012. "So many teams, so little time: Time allocation matters in geographically dispersed teams," *Journal of Organizational Behavior* (33:3), pp. 316–341.
Dahlander, L. and O'Mahony, S. 2011. "Progressing to the Center: Coordinating Project Work," *Organization Science* (22:4), pp. 961-979.
Daniel, S., Agarwal, R. and Stewart, K.J. 2013. "The Effects of Diversity in Global, Distributed Collectives: A Study of Open Source Project Success," *Information Systems Research* (24:2), pp. 312-333.
Darrow, J. B., and Behrend, T. S. 2017. *Person-environment fit is a formative construct. Journal of Vocational Behavior*, 103, pp. 117-131.
Deci, E. L. 1971. "The effects of externally mediated rewards on intrinsic motivation," Journal of Personality and Social Psychology, 18, pp. 105-115.
Deci, E.L., Koestner, R. and Ryan, R.M. 1999. "A meta-analytic review of experiments examining the effects of extrinsic rewards on intrinsic motivation," *Psychological Bulletin*, (125:6), pp. 627–668.
Dryden, A. 2013. "The ethics of unpaid labor and the OSS community," available at: https://www.ashedryden.com/blog/the-ethics-of-unpaid-labor-and-the-oss-community
Ducheneaut, N. 2005. "Socialization in an Open Source Software Community: A Socio-Technical Analysis," *Computer Supported Cooperative Work* (14:4), pp. 323-368.
Edwards, J. and Cable, D. 2009. "The Value of Value Congruence," *Journal of Applied Psychology* (94:3), pp. 654-677.

Ehls, D. 2017. "Open Source Project Collapse – Sources and Patterns of Failure*," Hawaii International Conference on System Sciences* Waikoloa, Hawaii, pp.
Ertas, N. 2019. Turnover intentions of volunteer resource managers: The roles of work motivations, person–organization fit, and emotional labor. *International Journal of Public Administration*, 429, pp. 741-752.
Fang, Y. and Neufeld, D. 2009. "Understanding Sustained Participation in Open Source Software Projects," *Journal of Management Information Systems* (25:4), pp. 9 - 50.
Fitzgerald, B. 2006. The transformation of open source software. *MIS quarterly*, pp. 587-598.
Foucault, M., Palyart, M., Blanc, X., Murphy, G.C. and Falleri, J.R. 2015. "Impact of developer turnover on quality in open-source software," In Proceedings of the *10th Joint Meeting on Foundations of Software Engineering* pp. 829-841.
Frederick, S., & Lowenstein, G. 1999. Hedonic Adaptation. In D. Kahneman, E. Diener, & N. Schwarz (Eds.), Well-being: The foundations of hedonic psychology. (pp. 302-329). New York: Russell Sage
Friedlander, F. and Walton, E. 1964. "Positive and negative motivations toward work," *Administrative Science Quarterly* pp. 194-207.
French, J. R. P., Jr., Rodgers, W. L. and Cobb, S. 1974. "Adjustment as person-environment fit," in Coelho, G., Hamburg, D. and Adams, J. (Eds) *Coping and Adaptation*, Basic Books, New York, pp. 316-333.
Frey, B. S., and Jegen, R. 2001. "Motivation crowding theory," *Journal of economic surveys*, (15:5), pp. 589-611.
Furnham, A., Forde, L. and Ferrari, K. 1999. "Personality and work motivation," *Personality and individual differences* (26:6), pp. 1035-1043.
Gamalielsson, J., and Lundell, B. 2014. "Sustainability of Open Source software communities beyond a fork: How and why has the LibreOffice project evolved?," *Journal of Systems and Software*, 89, pp. 128-145.
Gerlach, J.H., Wu, C.-G., Cunningham, L.F. and Young, C.E. 2016. "An Exploratory Study of Conflict over Paying Debian Developers," *International Journal of Open Source Software and Processes* (7:3), pp. 20-38.
Germonprez, M., Allen, J.P., Warner, B., Hill, J. and McClements, G. 2013. "Open source communities of competitors," *interactions* (20:6), pp. 54-59.
Germonprez, M., Kendall, J.E., Kendall, K.E., Mathiassen, L., Young, B. and Warner, B. 2016. "A theory of responsive design: A field study of corporate engagement with open source communities," *Information Systems Research* (28:1), pp. 64-83.
Ghosh, K., Ramakrishnan, T. and Khuntia, J. 2013. "Open Source Software and Performance: A Fit Perspective," *Pacific Asia Conference on Information Systems* pp. 116.
Hair, J. F., Black, W. C., Babin, B. J., and Anderson, R. E. 2009. Multivariate Data Analysis. 7th Edition. Pearson Prentice Hall.
Hahn, J., Moon, J. Y., and Zhang, C. 2008. "Emergence of new project teams from open source software developer networks: Impact of prior collaboration ties," *Information Systems Research*, (19:3), pp. 369-391.
Hansson, D.H. 2013. "The perils of mixing open source and money," available at: https://dhh.dk/2013/the-perils-of-mixing-open-source-and-money.html
Haring, M. J., Stock, W. A., and Okun, M. A. 1984. "A research synthesis of gender and social class as correlates of subjective well-being," *Human Relations*, 37, pp. 645-657.
Hars, A. and Ou, S. 2002. "Working for Free? Motivations for Participating in Open Source Projects," *International Journal of Electronic Commerce* (6:3), pp. 25-39.
Hartzel, J., Agresti, A. and Caffo, B. 2001. "Multinomial logit random effects models," *Statistical Modelling* (1:2), pp. 81-102.
Henkel, J. 2006. "Selective revealing in open innovation processes: The case of embedded Linux," *Research policy* (35:7), pp. 953-969.

Henkel, J. 2008. "Champions of revealing—the role of open source developers in commercial firms," *Industrial and Corporate Change* (18:3), pp. 435–471.
Hertel, G., Niedner, S. and Herrmann, S. 2003. "Motivation of software developers in Open Source projects: an Internet-based survey of contributors to the Linux kernel," *Research Policy* (32:7), pp. 1159-1177.
Herzberg, F. 1965. "The motivation to work among Finnish supervisors," *Personnel Psychology* (18:4), pp. 393-402.
Herzberg, F. 1968. "One more time: How do you motivate employees?," *Harvard Business Review,* pp. 53-62.
Herzberg, F., Snyderman, B.B. and Mausner, B. 1966. "*The Motivation to Work,"* Wiley.
Ho, S.Y. and Rai, A. 2017. "Continued Voluntary Participation Intention in Firm-Participating Open Source Software Projects " *Information Systems Research* (28:3), pp. 603-625.
Huckman, R. S., Staats, B. R., and Upton, D. M. 2009. "Team familiarity, role experience, and performance: Evidence from Indian software services," *Management science*, (55:1), pp. 85-100.
Inglehart, R. (1997). Modernization and post modernization: Cultural, economic and political change in societies. Princeton, NJ: Princeton University Press.
Jarczyk, O., Jaroszewicz, S., Wierzbicki, A., Pawlak, K. and Jankowski-Lorek, M. 2018. "Surgical teams on GitHub: Modeling performance of GitHub project development processes," *Information and Software Technology* pp.
Jensen, C. and Scacchi, W. 2005. "Collaboration, leadership, control, and conflict negotiation and the netbeans. org open source software development community*," 38th Annual Hawaii International Conference In System Sciences*, Big Island, HI, pp. 1-9.
Jensen, C. and Scacchi, W. 2007. "Role migration and advancement processes in ossd projects: A comparative case study," *29th International Conference on ICSE IEEE* pp.
Joyce, E. and Kraut, R.E. 2006. "Predicting continued participation in newsgroups*,"* Wiley Online Library.
Kantrowitz, N. 2015. Funding FOSS. Available at: https://coderanger.net/funding-foss/
Kasser, T. 2002. The high price of materialism. Cambridge, MA: MIT Press.
Katt, J.A. and Condly, S.J. 2009. "A Preliminary Study of Classroom Motivators and De-Motivators from a Motivation-Hygiene Perspective," *Communication Education* (58:2), pp. 213-234.
Ke, W. and Zhang, P. 2009. "Motivations in open source software communities: The mediating role of effort intensity and goal commitment," *International Journal of Electronic Commerce* (13:4), pp. 39-66.
Ke, W., & Zhang, P. 2010. "The effects of extrinsic motivations and satisfaction in open source software development," *Journal of the Association for Information Systems*, (11:12), pp. 784-808.
Kohn, A. 1999. "*Punished by Rewards:: The Trouble with Gold Stars, Incentive Plans, A's, Praise, and Other Bribes,"* Houghton Mifflin Harcourt.
Kreutzer, K., and Jäger, U. 2011. Volunteering versus managerialism: Conflict over organizational identity in voluntary associations. *Nonprofit and Voluntary Sector Quarterly*, 404, pp. 634-661.
Krishnamurthy, S. and Tripathi, A.K. 2009. "Monetary donations to an open source software platform," *Research Policy* (38:2), pp. 404-414.
Kristof-Brown, A., Zimmerman, R. and Johnson, E. 2005. "Consequences of individuals' fit at work: A meta-analysis of person-job, person-organization, person-group and person-supervisor fit," *Personnel Psychology* (58:2), pp. 281-342.
Kristof-Brown, A.L. and Guay, R.P. 2011. "*Person-Environment Fit,"* Washington D.C.: American Psychological Association.
Kristof, A.L. 1996. "Person organization fit: An integrative review of its conceptualizations, measurement, and implications," *Personnel Psychology* (49:1), pp. 1-49.
Lakhani, K. and Wolf, R. 2005. "*Why Hackers Do What They Do: Understanding Effort and Motivation in Open Source Projects,"* Cambridge, Mass: MIT Press.
Lauver, K.J. and Kristof-Brown, A. 2001. "Distinguishing between employees' perceptions of person–job

and person–organization fit," *Journal of vocational behavior* (59:3), pp. 454-470.
Lee, A. 2018. "One-Time Contributors to FLOSS: Surveys and Data Analysis," *ACM SIGSOFT Software Engineering Notes* (43:1), pp. 1-6.
Lee, G. and Cole, R. 2003. "From a Firm-Based to a Community-Based Model of Knowledge Creation: The Case of the Linux Kernel Development," *Organization Science* (14:6), pp. 633-649.
Lerner, J. and Tirole, J. 2005. "The scope of open source licensing," *Journal of Law, Economics, and Organization* (21:1), pp. 20-56.
Lesaffre, E. and Spiessens, B. 2001. "On the effect of the number of quadrature points in a logistic random effects model: an example," *Journal of the Royal Statistical Society: Series C (Applied Statistics)* (50:3), pp. 325-335.
Lewin, K. (1935). Dynamic theory of personality. New York: McGraw-Hill.
Lin, B., Robles, G. and Serebrenik, A. 2017. "Developer turnover in global, industrial Open Source projects: Insights from applying survival analysis," *Proceedings of the 12th International Conference on Global Software Engineering* pp. 66-75.
Lindberg, A., Berente, N., Gaskin, J. and Lyytinen, K. 2016. "Coordinating interdependencies in online communities: A study of an open source software project," *Information Systems Research*, (27:4), pp. 751-772
Linux Foundation. 2019. "Recruiting Open Source Developers, " available at: https://www.linuxfoundation.org/resources/open-source-guides/recruiting-open-source-developers/
Ljungberg, J. 2000. "Open source movements as a model for organizing," *European Journal of Information Systems* (9:4), pp. 208-216.
MacCormack, A., Rusnak, J. and Baldwin, C.Y. 2006. "Exploring the Structure of Complex Software Designs: An Empirical Study of Open Source and Proprietary Code," *Management Science* (52:7), pp. 1015-1031.
Marquess, S. 2014. "Of Money, Responsibility, and Pride," available at: http://veridicalsystems.com/blog/of-money-responsibility-and-pride/
Maruping, L. M., Daniel, S. L., & Cataldo, M. 2019. "Developer centrality and the impact of value congruence and incongruence on commitment and code contribution activity in open source software communities," *MIS Quarterly*, (43:3), pp. 951-976.
Moore, G.C. and Benbasat, I. 1991. "Development of an Instrument to Measure the Perceptions of Adopting an Information Technology Innovation," *Information Systems Research* (2:3), pp. 192-222.
Moreland, R. L. 1999. Transactive memory: Learning who knows what in work groups and organizations. In L. L. Thompson, J. M. Levine, & D. M. Messick (Eds.), Shared cognition in organizations, pp. 3–31. Mahwah, NJ: Erlbaum.
Moreland, R.L. and Levine, J.M. 1982. "Socialization in small groups: Temporal changes in individual-group relations," *Advances in experimental social psychology* (15: pp. 137-192.
Moynihan, D. P., and Pandey, S. K. 2008. The ties that bind: Social networks, person-organization value fit, and turnover intention. *Journal of public administration research and theory*, 182, pp. 205-227.
Muchinsky, P. M., and Monahan, C. J. 1987. "What is person–environment congruence? Supplementary versus complementary models of fit," *Journal of Vocational Behavior,* 31, pp. 268−277.
O'Leary, M.B., Mortensen, M. and Woolley, A.W. 2011. "Multiple Team Membership: A Theoretical Model of its Effects on Productivity and Learning for Individuals and Teams," *Academy of Management Review* (36:3), pp. 461-478.
O'Mahony, S. and Ferraro, F., 2007. "The emergence of governance in an open source community," *Academy of Management Journal* (50:5), pp.1079-1106.
Oh, W., Moon, J.Y., Hahn, J. and Kim, T. 2016. "Leader Influence on Sustained Participation in Online Collaborative Work Communities: A Simulation-Based Approach," *Information Systems Research* pp.

Palyart, M., Murphy, G.C. and Masrani, V. 2017. "A Study of Social Interactions in Open Source Component Use," *IEEE Transactions on Software Engineering* pp.

Petter, S., Straub, D. and Rai, A. 2007. "Specifying formative constructs in information systems research," *Mis Quarterly* pp. 623-656.

Pinheiro, J.C. and Chao, E.C. 2006. "Efficient Laplacian and adaptive Gaussian quadrature algorithms for multilevel generalized linear mixed models," *Journal of Computational and Graphical Statistics* (15:1), pp. 58-81.

Pinto, G., Steinmacher, I. and Aur, M. 2016. "More Common Than You Think: An In-Depth Study of Casual Contributors*," IEEE 23rd International Conference on Software Analysis, Evolution, and Reengineering (SANER)*, Osaka, Japan, pp. 112-123.

Podsakoff, P.M., MacKenzie, S.B., Lee, J.-Y. and Podsakoff, N.P. 2003. "Common method biases in behavioral research: a critical review of the literature and recommended remedies," *Journal of Applied Psychology* (88:5), pp. 879-903.

Qureshi, I. and Fang, Y. 2010. "Socialization in Open Source Software Projects: A Growth Mixture Modeling Approach," *Organizational Research Methods* (14:1), pp. 208-238.

Raudenbush, S.W. and Bryk, A.S. 2002. "*Hierarchical linear models: Applications and data analysis methods,"* Sage.

Raymond E. 2001. The cathedral and the Bazaar. Musings on Linux and Open Source by an Accidental Revolutionary. O'Reilly and Associates, Sebastopolous, C.A., USA.

Riehle, D., Philipp Riemer, Kolassa, C. and Schmidt, M. 2014. "Paid vs. volunteer work in open source," *47th Hawaii International Conference on System Sciences* 3286-3295.

Robles, G., and Gonzalez-Barahona, J.M. 2003. "Contributor Turnover in Libre Software Projects," in: *IFIP International Federation for Information Processing: Open Source Systems*, Springer, Boston, pp 273-286.

Federation for Information Processing: Open Source Systems, Springer, Boston, 2006.

Roberts, J.A., Hann, I.-H. and Slaughter, S.A. 2006. "Understanding the Motivations, Participation, and Performance of Open Source Software Developers: A Longitudinal Study of the Apache Projects," *Management Science* (52:7), pp. 984-1000.

Ryan, R. M., and Deci, E. L. 2000. "Self-determination theory and the facilitation of intrinsic motivation, social development, and well-being," *American psychologist*, (55:1), pp. 68-78.

Sachau, D.A. 2007. "Resurrecting the motivation-hygiene theory: Herzberg and the positive psychology movement," *Human resource development review* (6:4), pp. 377-393.

Schilling, A., Laumer, S. and Weitzel, T. 2012. "Who Will Remain? - An Evaluation of Actual Person-Job and Person-Team Fit to Predict Developer Retention in FLOSS Projects*," 45th Hawaii International Conference on System Sciences*, Maui, pp. 3446-3455.

Schlueter, I. 2013. "Money and Open Source," *Medium*, available at: https://medium.com/open-source-life/money-and-open-source-d44a1953749c

Schneider, B. 1987. "The people make the place," *Personnel psychology*, (40:3), pp. 437-453.

Schwartz, S. 1992. Universals in the content and structure of values: Theoretical advances and empirical tests in 20 countries. In M. P. Zanna (Ed.). Advances in experimental social psychology, 25, pp. 1–65. San Diego, CA: Academic Press

Setia, P., Rajagopalan, B., Sambamurthy, V. and Calantone, R. 2012. "How Peripheral Developers Contribute to Open-Source Software Development," *Information Systems Research* (23:1), pp. 144-163.

Shah, S. 2006. "Motivation, Governance and the Viability of Hybrid Forms in Open Source Software Development," *Management Science* (52:7), pp. 1000-1014.

Singh, P. and Phelps, C. 2013. "Networks, Social Influence, and the Choice Among Competing Innovations: Insights from Open Source Software Licenses," *Information Systems Research* (24:3), pp. 539-560.

Solinger, O.N., Olffen, W.v., Roe, R.A. and Hofmans, J. 2013. "On Becoming (Un) Committed: A Taxonomy and Test of Newcomer Onboarding Scenarios," *Organization Science* (24:6), pp.

1640-1661.
Spaeth, S., von Krogh, G. and He, F. 2014. "Research Note—Perceived Firm Attributes and Intrinsic Motivation in Sponsored Open Source Software Projects," *Information Systems Research* (26:1), pp. 224-237.
Steinmacher, I., Conte, T., Gerosa, M.A. and Redmiles, D. 2015. "Social barriers faced by newcomers placing their first contribution in open source software projects," *Proceedings of the 18th ACM conference on Computer Supported Cooperative Work & Social Computing*, Vancouver, BC, Canada, pp. 1379-1392.
Stewart, K. and Gosain, S. 2006. "The Impact of Ideology on Effectiveness in Open Source Software Development Teams," *Management Information Systems Quarterly* (30:2), pp. 291-314.
Stewart, K. J., Ammeter, A. P., & Maruping, L. M. 2006. "Impacts of license choice and organizational sponsorship on user interest and development activity in open source software projects," *Information Systems Research*, (17:2), pp. 126-144.
Van Vianen, A. E., Pater, I. E. D., and Dijk, F. V. 2007. Work value fit and turnover intention: same-source or different-source fit. *Journal of managerial psychology*, 222, pp. 188-202.
Von Krogh, G., Haefliger, S., Spaeth, S. and Wallin, M.W. 2012. "Carrots and Rainbows: Motivation and Social Practice in Open Source Software Development," *MIS Quarterly* (36:2), pp. 649-676.
Von Krogh, G., Spaeth, S. and Lakhani, K. 2003. "Community, joining, and specialization in open source software innovation: a case study," *Research Policy* (32:7), pp. 1217-1241.
Wang, T., Zhang, Y., Yin, G., Yu, Y. and Wang, H. 2018. "Who Will Become a Long-Term Contributor?: A Prediction Model based on the Early Phase Behaviors," *Proceedings of the Tenth Asia-Pacific Symposium on Internetware*.
Werdmuller, B. 2017. "Why open source software isn't as ethical as you think it is (And why it is)," available at: https://words.werd.io/why-open-source-software-isnt-as-ethical-as-you-think-it-is-2e34d85c3b16
Widaman, K.F. 2007. Common factors versus components: Principals and principles, errors and misconceptions. Cudeck, R. and MacCallum, RC eds. Factor Analysis at 100: Historical Developments and Future Direction, pp. 177-203.
Wiersma, U. J. 1992. The effects of extrinsic rewards in intrinsic motivation: A meta-analysis," *Journal of Occupational & Organizational Psychology*, (65:2), pp. 101-114.
Willett, J. B. 1997. "Measuring change: What individual growth modeling buys you," *Change and development: Issues of theory, method, and application*, pp. 213-243.
Xiao, X., Lindberg, A., Hansen, S. and Lyytinen, K. 2018. "'Computing' Requirements for Open Source Software: A Distributed Cognitive Approach," *Journal of the Association for Information Systems* pp.
Xu, B., Jones, D.R. and Shao, B. 2009. "Volunteers' involvement in online community based software development," *Information and Management* (46:3), pp. 151-158.
Zhou, M., Mockus, A., Ma, X., Zhang, L. and Mei, H. 2016. "Inflow and Retention in OSS Communities with Commercial Involvement: A Case Study of Three Hybrid Projects," *ACM Transactions on Software Engineering and Methodology* (25:2), pp. 1-29.

Appendix – A
Survey Items

<table>
<tr><th>Construct</th><th>Item</th><th>Survey Question</th><th>Source</th></tr>
<tr><td rowspan="6">Value Fit</td><td>Practice</td><td>My personal beliefs about open source practices match well with the project.</td><td rowspan="6">Measures the match between developer and projects values, norms, and beliefs (Cable and DeRue 2002).

OSS Specific Factors:
• License restrictiveness and market sponsorship (Stewart et al. 2006)
• Forking norms and named credit policy norms (Stewart and Gosain 2006)
• Reciprocation norms (Shah 2006).
• Beliefs regarding OSS practices (Stewart 2006)</td></tr>
<tr><td>Forking</td><td>The project's norms about when and how to fork are a good fit with my personal norms.</td></tr>
<tr><td>Credit</td><td>The project's policy about giving due credit to developers for their contributions matches well with my personal norms.</td></tr>
<tr><td>License</td><td>My personal values about open source software usage and sharing match the project's license.</td></tr>
<tr><td>Reciprocate</td><td>My personal norms about reciprocating favors match well with the project's norms.</td></tr>
<tr><td>Sponsorship</td><td>My personal values about external sponsorships for supporting open-source projects match well with the external sponsorship(s) that support this project.</td></tr>
<tr><td rowspan="7">Need-Supply Fit</td><td>Career</td><td>There is a good fit between my need to enhance my career opportunities and the opportunities that the project provides.</td><td rowspan="7">Measures the match between the developer's needs and what the project supplies (Cable and DeRue 2002).

OSS Specific Factors:
• Enjoyment/fun in programming and Peer Recognition (Shah 2006).
• Feeling of competence (Hars and Ou 2002).
• Learning
• Personal/professional use of software
• Self-marketing (Career Enhancement).
• Financial Compensation (Hars and Ou 2002).</td></tr>
<tr><td>Financial</td><td>There is a good match between my expectation of financial compensation and what contributing to this project provides me.</td></tr>
<tr><td>Learning</td><td>There is a good fit between the learning opportunities the project provides me and what I expect by contributing to it.</td></tr>
<tr><td>Recognition</td><td>There is a good match between my desire for recognition for a job well done and what the project offers me.</td></tr>
<tr><td>Enjoy</td><td>There is a good fit between the enjoyment I seek while contributing to the project and what it provides.</td></tr>
<tr><td>Competent</td><td>There is a good match between my need to feel competent and what participating in the project offers me.</td></tr>
<tr><td>Software*</td><td>There is a good fit between my need for software for personal or professional use, and what the project provides me.</td></tr>
<tr><td rowspan="3">Demand-Ability Fit</td><td>Knowledge</td><td>There is a good fit between my knowledge and the demands of the project.</td><td rowspan="3">Measures the match between the project's demands and the abilities of the developer to meet those demands (Cable and DeRue 2002).

OSS Specific Factors:
Typically measured along three dimensions (KSA): Knowledge, Skills, and Abilities (Caldwell & O'Reilly, 1990).</td></tr>
<tr><td>Skills</td><td>My skills match well with the requirements of the project.</td></tr>
<tr><td>Abilities</td><td>My abilities are well matched with the demands that the project places on me.</td></tr>
</table>

* Based on the rotated component matrix results in the PCA analysis, the Software item was dropped from further consideration.

Appendix-B
Panel B: Mixed Model for Focal Commits

-------------------------------------INTERCEPTS-------------------------------------

Code Commit Level Intercept at t=3 Months

γ_{000}

Growth Trajectory Intercept

+ γ_{100}*Month*

-------------------------------------COVARIATES-------------------------------------

Time-Varying Covariates and Errors

+ γ_{200}*Non-Focal Commits* + γ_{300}*Non-Focal Pulls* + γ_{400}*Non-Focal Issues* + r_0 + r_1*Month*+ u_{00} + *e*

Code Commit Level Developer Covariates

+ γ_{020}*Tenure* + γ_{030}*Past Commits in Focal Project* + γ_{040}*Number of Project Associations* + γ_{050}*Followers* + γ_{070}*Need-Supply Fit* + γ_{080}*Demand-Ability Fit*

Code Commit Level Project Covariates

+ γ_{001}*Project Age* + γ_{002}*Ln (Commits) in Focal Project* + γ_{003}*Releases* + γ_{004}*Contributors in Focal Project* + γ_{005}*Ln (Stars) in Focal Project* + γ_{006}*Ln (Forks) in Focal Project*

Growth Trajectory Developer Covariates

+ γ_{120}*Month*Tenure* + γ_{130}*Month*Past Commits in Focal Project* + γ_{140}*Month*Number of Project Associations* + γ_{150}*Month*Followers* + γ_{170}*Month*Need-Supply Fit* + γ_{180}*Month*Demand-Ability Fit*

Growth Trajectory Project Covariates

+ γ_{101}*Month*Project Age* + γ_{102}*Month*Ln (Commits) in Focal Project)* + γ_{103}*Month*Releases* + γ_{104}*Month*Contributors in Focal Project* + γ_{105}*Month*Ln (Stars) in Focal Project* + γ_{106}*Month*Ln (Forks) in Focal Project*

-------------------------------------PREDICTORS-------------------------------------

Code Commit Level Predictors

+ γ_{010}*Paid* + γ_{060}*Value Fit* + γ_{090}*Paid*Value Fit*

Growth Trajectory Predictors

+ γ_{110}*Month*Paid* + γ_{160}*Month* Value Fit* + γ_{190}*Month*Paid* Value Fit*

Appendix-C
Robustness Checks

Variable Type	Fixed Effects	4 Month Model	Unit-specific Model	Quadratic Model
Mean Code Commit Level	Intercept (π_0)	2.81*** (0.03)	0.11*** (0.01)	2.86*** (0.03)
Mean Growth Rate	Month (π_1)	0.11*** (0.01)	-0.21*** (0.00)	-0.16*** (0.02)
Acceleration Curve	Month^2 (π_2)			0.04*** (0.00)
Time Varying Covariates				
Level 1 Control	Non-Focal Commits (π_2)	-0.00*** (0.00)	-0.00*** (0.00)	-0.00*** (0.00)
	Non-Focal Pulls (π_3)	0.00 (0.00)	0.01*** (0.00)	0.01*** (0.00)
	Non-Focal Issues (π_4)	0.01*** (0.00)	0.00*** (0.00)	0.01*** (0.00)
Code Commit Level Covariates (at t=3 months)				
Developer Control (Level 2)	Tenure (β_{02})	0.06*** (0.02)	0.08*** (0.01)	0.12*** (0.02)
	Past Commits in Focal Project (β_{03})	0.00*** (0.00)	0.00*** (0.00)	0.00*** (0.00)
	Project Associations (β_{04})	-0.00*** (0.00)	-0.00 (0.00)	-0.00 (0.00)
	Followers (β_{05})	0.00*** (0.00)	0.00 (0.00)	0.00 (0.00)
	Need-Supply Fit (β_{07})	0.36*** (0.03)	0.39*** (0.02)	0.36*** (0.02)
	Demand-Ability Fit (β_{08})	0.46*** (0.02)	0.44*** (0.02)	0.42*** (0.02)
Project Control (Level 3)	Project Age (γ_{001})	0.06** (0.02)	0.07*** (0.00)	0.07** (0.02)
	LnCommits (γ_{002})	0.19*** (0.02)	0.19*** (0.00)	0.21*** (0.01)
	Releases (γ_{003})	-0.00 (0.00)	-0.00*** (0.00)	-0.00** (0.00)
	Contributors (Size) (γ_{004})	-0.00*** (0.00)	-0.00***(0.00)	-0.00*** (0.00)
	LnStars (γ_{005})	0.20*** (0.03)	0.17*** (0.01)	0.19*** (0.02)
	LnForks (γ_{006})	-0.12*** (0.04)	-0.08*** (0.01)	-0.11*** (0.03)
Growth Trajectory Covariates (Change Over Time)				
Developer Control (Level 2)	Tenure (β_{12})	0.09*** (0.01)	0.06*** (0.00)	0.06*** (0.00)
	Past Commits in Focal Project (β_{13})	-0.00** (0.00)	-0.00** (0.00)	-0.00** (0.00)
	Project Associations (β_{14})	0.00*** (0.00)	0.00*** (0.00)	0.00*** (0.00)
	Followers (β_{15})	-0.00 (0.00)	-0.00*** (0.00)	-0.00*** (0.00)
	Need-Supply Fit (β_{17})	0.06*** (0.01)	0.01 (0.01)	0.03*** (0.00)
	Demand-Ability Fit (β_{18})	-0.00 (0.01)	-0.00 (0.01)	-0.00 (0.01)
Project Control (Level 3)	Project Age (γ_{101})	0.02* (0.01)	0.02*** (0.00)	0.02*** (0.00)
	LnCommits (γ_{102})	0.00 (0.01)	0.00 (0.00)	0.00 (0.00)
	Releases (γ_{103})	0.00*** (0.00)	-0.00*** (0.00)	0.00*** (0.00)
	Contributors (Size) (γ_{104})	0.00 (0.00)	-0.00*** (0.00)	0.00 (0.00)
	LnStars (γ_{105})	0.00* (0.00)	0.00 (0.00)	0.00 (0.01)
	LnForks (γ_{106})	-0.00 (0.01)	-0.00 (0.00)	-0.01* (0.00)
Code Commit Level Predictors (at t=3 months)				
Predictor of Developer Code Commit Level	Paid (β_{01})	0.67*** (0.05)	0.73*** (0.01)	0.69*** (0.05)
	Value Fit (β_{06})	0.03 (0.02)	0.10*** (0.02)	0.09*** (0.02)
	Paid X Value Fit (β_{09})	0.16** (0.07)	0.12*** (0.03)	0.06 (0.07)
Growth Trajectory Predictors (Change Over Time)				

Predictor of Developer Growth Trajectory	Paid (β_{11})	0.03** (0.02)	0.03*** (0.01)	0.03** (0.01)
	Value Fit (β_{16})	0.27*** (0.01)	0.09*** (0.01)	0.10*** (0.00)
	Paid X Value Fit (β_{19})	-0.30*** (0.01)	-0.12*** (0.02)	-0.12*** (0.01)

Notes: 0.10; ** 0.05; *** 0.01; Robust Standard Errors are in parentheses.

Author Bios

Pratyush Nidhi Sharma is an Assistant Professor in the Department of Information Systems, Statistics and Management Science, in the University of Alabama's Culverhouse College of Business. Pratyush holds a PhD in Information Systems from the University of Pittsburgh. His research interests are interdisciplinary with a focus on the development, application, and the impact of Information Systems. On the technology supply side, his work investigates how online collaboration communities can better develop technological artifacts such as open-source software. On the demand side, he investigates factors that affect the adoption of IT, and the effect on user satisfaction and firm performance. In addition, he focuses on developing predictive-analytic tools to better utilize the strengths of the prediction-oriented approach (vis-à-vis the explanation-oriented approach) to create robust theory and policy. He has published in distinguished management research journals including the Journal of the Association for Information Systems, Journal of Retailing, Decision Sciences, AIS Transactions on Human-Computer Interaction, Government Information Quarterly, and Journal of Information Systems.

Sherae L. Daniel is an Associate Professor of Operations, Business Analytics and Information Systems in the Carl H. Lindner College of Business at the University of Cincinnati. She earned her Ph.D. in Information Systems from the Robert H. Smith School of Business at the University of Maryland. Sherae's research seeks to reveal how to best manage collaboration challenges in nontraditional work environments. In particular, she seeks to uncover the keys that will unlock doors to future success for OSS collaborators. Sherae's research has been published in premier outlets such as Information Systems Research, MIS Quarterly, and the Journal of Association for Information Systems. She is a member of the Association for Information Systems.

Varun Grover is the David Glass Endowed Chair and Distinguished Professor of IS at the Walton School of Business, University of Arkansas. His current work focuses on the impacts of digitalization on individuals and organizations. Dr. Grover has published extensively in the information systems field, with over 250 publications in major refereed journals with ten recent articles ranking him among the top four researchers globally. Dr. Grover has an h-index of 93 which is ranked in the top 5 in the field, and over 40,000 citations in Google Scholar. Thompson Reuters recognized him as one of 100 Highly Cited Scholars globally in all Business disciplines. He is has held or currently holds senior editorial positions in many top IS journals including MISQ, JAIS, MISQE, ISR, etc. He is recipient of numerous awards from USC, Clemson, University of Arkansas, AIS, Academy of Management, DSI, the OR Society, Anbar, PriceWaterhouse, among others for his research and teaching. He is a Fellow of the Association for Information Systems and was recently recognized with the LEO Award for exceptional lifetime accomplishment in IS.

Dr. Tingting (Rachel) Chung is Clinical Associate Professor of Operations and Information Systems. She holds a Ph.D. in Business Administration/Management Information Systems, a

Ph.D. in Psychology, and a Master of Science in Information Science, all from the University of Pittsburgh. Dr. Chung's research has been published in Journal of Association for Information Systems, Communications of the ACM, Journal of Managerial Psychology, International Journal of Production Economics, Journal of Information & Knowledge Management, AIS Transactions on HCI, and Omega. She has also given numerous presentations at international conferences, including ICIS, ICAIF, AoM, INFORMS, and SIGCSE. Dr. Chung's research has been supported by ACFE Research Institute (ARI), Blockchain Lab of William & Mary, and National Security Agency. Dr. Chung has completed a visiting scholarship at Vietnam National University - International School and has received IBM Faculty Award in the Cognitive Computing category, Community Partner of the Year Award, Houston Methodist Hospital, on behalf of INFORMS, and Faculty Excellence Award from the Master of Science in Business Analytics (MSBA) program at College of William & Mary.